On the Nonlinear Dynamics of Convective-Diffusive-Reactive Waves

Loizos Christodoulou

Submitted in fulfilment of the requirements for the
Degree of Doctor of Philosophy in Mechanical Engineering

School of Engineering
College of Science and Engineering
University of Glasgow

September 2018

Abstract

The burning of fossil fuels will play a crucial role for both power generation and transportation sectors over the next fifty years according to the forecasts of regulatory bodies in these sectors. The combustion of these fuels contributes to climate change, air pollution, and environmental noise and consequently affects the health and quality of human life. Public awareness and governmental legislations over these issues have improved substantially in recent years. As a result, heat engine manufacturers have to meet international standards for noise and pollutant emissions which are continually lowering the acceptable levels.

The manufacturers of gas turbine engines which are used widely for propulsion and power generation are investing continually into extensive research that will enable them to meet the ever more stringent targets of NO_x emissions and perceived noise. It is widely acknowledged among gas turbine manufacturers that lean premixed combustors fuelled by hydrogen blends is the most effective solution. The burning of fuels containing hydrogen results in reductions of all emissions compared to the levels produced from the burning of conventional fuels with the exception of NO_x. Significant reductions of NO_x emissions is the primary role of lean premixed combustion systems. These systems also provide reduced fuel consumption that results in less CO_2 and H_2O production because they operate close to the fuel-lean blowout limit. That is, these systems burn a fuel-air ratio that is lower than stoichiometry.

Lean premixed combustors are effective in achieving low NO_x emissions but their operating range is severely limited due to their susceptibility to a range of dynamical problems including combustion instabilities and flame flashback. Furthermore, these combustors are more noisy compared to other combustion systems for two reasons. Firstly, the flame dynamics are more unsteady and hence, more energy is supplied to the acoustic field. Secondly, the cooling air that flows around the combustor liner and acts as an acoustic insulation layer is relatively much less. The requirement for fuel flexibility makes these problems more challenging to solve as the chemical composition of the fuel significantly influences the combustion dynamics.

Combustion instabilities are characterised by strong pressure oscillations that can destroy the combustor. These are a result of coupling of the flame heat release with flow perturbations through a feedback mechanism that is usually the combustor acoustics. In the multiphysics environment of a combustor there are various processes that cause flow perturbations and hence, the problem of combustion instability is extremely complex. The flow perturbations could be

the result of other dynamic phenomena such as flame flashback and advecting entropy waves and these are the subject of the current work. The former is a highly transient phenomenon characterised by the sudden upstream propagation of the flame. The entropy waves are hot parcels of fluid that are generated by an oscillatory heat release at the flame and advect with the flow to generate acoustic waves known as entropy noise during their passage through the combustor exit nozzle. The flashback frequency of a flame periodically moving upstream or the frequency of entropy noise could coincide with an acoustic mode of the combustor, thus resulting in combustion instability. Even if the phenomena do not cause combustion instability they are problems in their own right. Flame flashback can damage upstream components of the combustor that are not designed to operate at high temperatures. On the other hand, entropy noise contributes to engine noise.

Investigation of flashback requires multiple, simultaneous diagnostics without prior knowledge of the relevant time and length scales of the physical processes involved. Detection, accordingly, deals with post-event characterization. The current work, attempts to detect subtle dynamics prior to flashback using for the first time nonlinear time series analysis tools to process existing pressure time series from flashback experiments. Time and frequency domain methods are unable to detect precursors of flashback as will be demonstrated. However, these conventional methods of time series analysis operate on the assumption that the source of the time series is linear. The highly transient nature of flashback clearly indicates that the phenomenon is a consequence of nonlinear dynamics. Following standard nonlinear time series analysis, the trajectory of the system in phase space is constructed from the time series data. Subsequently, the orbit of the trajectory is analysed using a running window to plot its translation error and recurrence quantification measures of its recurrence pattern as a function of time. The translation error analysis is applied to time series from a flashback experiment in stable combustion. The recurrence analysis is applied to time series from a flashback experiment in a different burner with unstable combustion. In both cases, it is found that the determinism of the system dynamics gradually increases as flashback is approached.

The influence of entropy noise on combustion instabilities is still a subject of contention. Experimental investigations are again difficult because entropy noise cannot be distinguished from noise generated at the flame in acoustic measurements. Hence, experimental investigations rely on measuring the temperature perturbations before their passage through the combustor exit nozzle. However, this brings about another difficulty, that of measuring high frequency temperature oscillations. Nonetheless, once the temperature measurements are made there is a need for a comprehensive theory to convert them to entropy noise. For this reason, research in this area has focused mainly on the acoustic response of nozzles to entropic forcing. The equally important stage of the phenomenon, that of the attenuation of advecting entropy waves through the combustor flow field has received little attention. Existing low order models of an advecting entropy wave are one-dimensional, linear, and purely phenomenological. The current

work, carries out a direct numerical simulation of an entropy wave advecting in a compressible turbulent channel flow with adiabatic and convectively cooled walls. Time series from the direct numerical simulation are subsequently processed using a novel methodology to develop a model that retains the two-dimensional shape and amplitude of the entropy wave. The model is capable of simulating the adiabatic and heat transferring cases using only a small fraction of the data from the direct numerical simulation. It is shown that a nonlinear model is more appropriate even for entropy waves with an amplitude that until now has been considered small. Also, it is found that heat loss at the walls significantly influences the advecting entropy wave.

Contents

List of Tables

List of Figures

Acknowledgements

I am especially grateful to my principal supervisor Dr. Nader Karimi for giving me the opportunity to undertake this work. I thank him for his invaluable guidance, personal encouragement, and continual support throughout my time as a postgraduate research student.

I would like to thank my additional supervisors Dr. Lipika Kabiraj, Dr. Andrea Cammarano, and Dr. Manosh Paul for their invaluable guidance and support during their stage of involvement. I extend my thanks to the Chair of Fluid Dynamics at the Technische Universität Berlin for hosting my research stay in the first year of my PhD study.

I gratefully acknowledge the full PhD studentship awarded to me by the Engineering and Physical Sciences Research Council (EPSRC). Moreover, I gratefully acknowledge the financial support from the Deutsche Forschungsgemeinschaft grant KA 3968/1-1 and the Systems, Power, and Energy (SPE) research division at the University of Glasgow.

Finally, I acknowledge that this work used the EPSRC funded high performance computing (HPC) centres Cirrus (http://www.cirrus.ac.uk) and ARCHIE-WeSt (www.archie-west.ac.uk). The later facility is funded by EPSRC grant no. EP/K000586/1.

Declaration

I am aware of and understand the policy of the University of Glasgow regarding plagiarism and I certify that the work in this thesis is my own, except where indicated by referencing, and is in accordance with University and School guidance on good academic conduct.

Publications stemming from this work

Pending review

L. Christodoulou and N. Karimi and A. Cammarano and M. Paul and S. Navarro-Martinez. State prediction of an entropy wave advecting through a turbulent channel flow. *Journal of Fluid Mechanics*, 2019.

Journal papers

L. Christodoulou and L. Kabiraj and A. Saurabh and N. Karimi. Characterizing the signature of flame flashback precursor through recurrence analysis. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 26(1):013110,2016.

Conference papers

L. Christodoulou and L. Kabiraj and A. Saurabh and J. Weinkauff and A. Dreizler and N. Karimi. Detection of flame flashback precursor using the translation error method. In *Proceedings of the 10th Asia-Pacific Conference on Combustion*, Beijing, China, 2015.

Chapter 1

Introduction

1.1 Motivation

The projections of the World Energy Council published in "The World Energy Scenarios: Composing energy futures to 2050" show that until the year 2050 fossil fuels will play a crucial role for both power generation and transportation sectors [54]. However, the activities of the above sectors contribute to climate change, air pollution, and environmental noise and consequently affect the health and quality of human life. Public awareness and governmental legislations over these issues have improved substantially in recent years. As a result, heat engine manufacturers have to meet international standards for noise and pollutant emissions which are continually lowering the acceptable levels.

Currently, gas turbine engines are used widely, either in stand-alone applications or combined with other power generation equipment and have substantially low emissions compared to other fossil-powered generation technologies [32]. Emission control technology for gas turbine engines has advanced dramatically over the past 30 years [108]. Nonetheless, manufacturers of gas turbine engines are continually striving to meet more stringent standards regarding pollutant emissions and noise by the year 2050. The Advisory Council for Aeronautics Research in Europe has set a target of reducing aviation NO_x emissions by 90% and perceived noise by 65% relative to levels in the year 2000 by the year 2050 [31].

The exhaust from gas turbines contains carbon dioxide (CO_2), water vapour (H_2O), carbon monoxide (CO), unburned hydrocarbons (UHC), particulate matter (smoke, ash, ambient non-combustibles, corrosion products, etc.), oxides of sulfur (SO_x), and oxides of nitrogen (NO_x) [141, 108]. Emissions from a typical two-engine jet aircraft during a 1-hour flight with 150 passengers are shown in figure 1.1a. These emissions pose a problem for both the environment and public health. In particular, the CO_2, H_2O, and NO_x contribute to global warming through the greenhouse effect. The CO and UHC are toxic and the latter combine with the NO_x to form photochemical smog. The NO_x and SO_x react with water molecules (H_2O) in the atmosphere to form acids that cause acid rain. The NO_x and soot have been shown to cause respiratory diseases

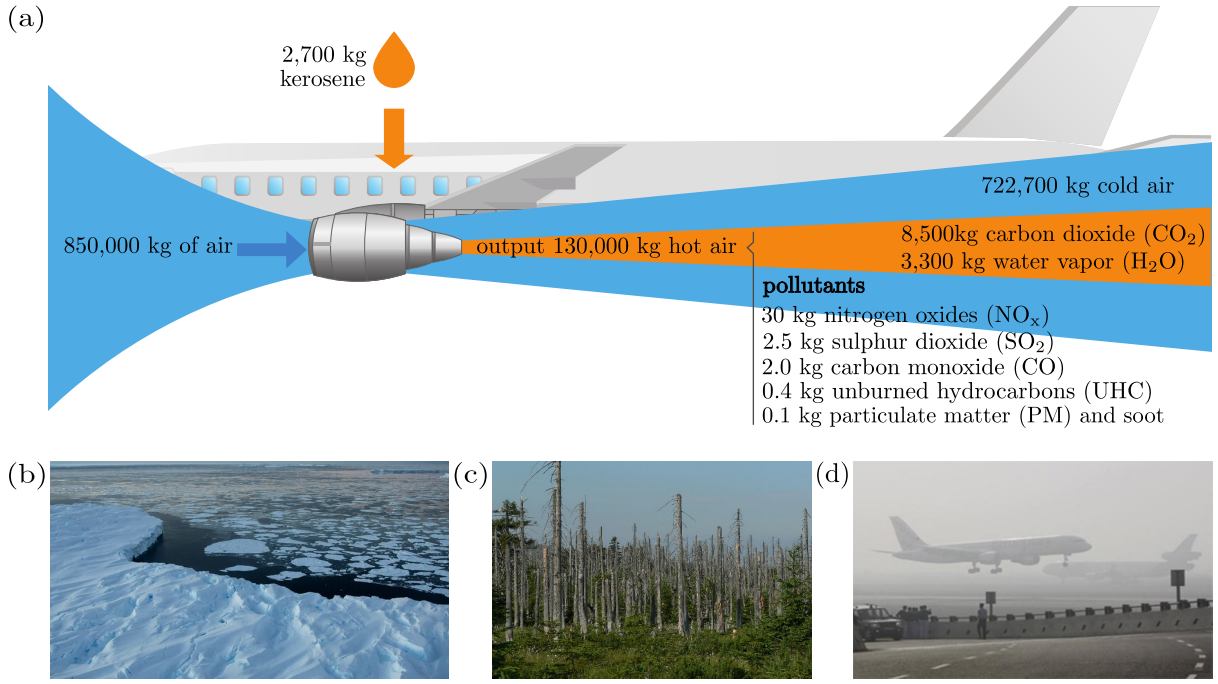

Figure 1.1: Combustion of fossil fuels: (a) Emissions from a typical two-engine jet aircraft during a 1-hour flight with 150 passengers [101], (b) Melting ice in Antarctica in 2016 due to the greenhouse effect, (c) Forestation in the Jizera mountains in 2006 destroyed due to acid rain, and (d) Delhi airport in 2017 engulfed in smog.

and problems. Examples of the impact of global warming, acid rain, and particulate matter on the environment and human activities around the world are shown in figure 1.1b-d.

The combustor is where emissions and significant noise are produced and therefore significant research and development effort is going into combustion technology. Emission controls for gas turbine engines were initially introduced in the early 1970s mainly for the reduction of NO_x. The simplest way of controlling NO_x emissions is by reducing the flame temperature in the combustor [108]. Thus, the early approach involved the injection of diluents such as water or steam directly into the primary combustion zone. The injection of water or steam into the combustor had minimal detrimental impact to the gas turbine cycle performance and the operational life of parts as long as the amount of water or steam injected was kept below a threshold [141]. The limitation of wet low NO_x technology was overcome with the advent of Dry Low NO_x (DLN) or Dry Low Emissions (DLE) combustion technologies, particularly that of lean premixed combustion in the late 1980s. Since then manufacturers have achieved significant reductions of the regulated emissions. Further reduction of emissions could be achieved by switching to fuels that contain more hydrogen than conventional fuels. Syngas [161, 23] and blends of hydrogen with gaseous hydrocarbons [16, 150, 170] are the main alternatives being considered.

Lean premixed combustion provides an important reduction of NO_x emissions with the added benefit of reduced fuel consumption, which results in less CO_2 and H_2O production. However, systems that utilize this type of combustion technology tend to be prone to a range

of dynamical problems including combustion instabilities and flame flashback that involve dynamics that are not yet fully understood [143]. Furthermore, the requirement for lean premixed systems to be fuel flexible makes these problems all the more challenging because the chemical composition of the fuel influences the combustion dynamics [85, 116]. The safe operation of lean premixed combustion systems is therefore limited to a narrow operational range. Consequently, the technology has only found application in industrial and not in aircraft gas turbines that are required to operate over a wide range of conditions. Moreover, lean premixed combustors are more noisy compared to other combustor technologies [39]. The reason for this is twofold. Firstly, the combustion process is more unsteady and hence, more energy is supplied to the acoustic field. Secondly, the cooling air that flows around the combustor liner and behaves as an acoustic insulation layer is much less as most of the air is supplied to the burner.

The operational and design constraints posed on lean premixed combustion systems by combustion instabilities and flame flashback on one hand and the comparable noise of combustion with jet and fan noise on the other hand have made the underlying dynamics of these and other combustion phenomena the focal point of extensive research [90, 20, 115, 143, 39, 131].

1.2 Formation mechanisms of oxides of nitrogen

The term NO_x is used to denote nitric oxide (NO) and nitrous oxide (NO_2) but in most flames NO is formed and then oxidises to NO_2 after the combustion products are released into the atmosphere [25, 166]. The formation of NO_x in the combustion of fuels that do not contain nitrogen occurs through three main mechanisms. These are the thermal or Zeldovich mechanism, the intermediate nitrous oxide mechanism and the prompt or Fenimore mechanism. In the combustion of fuels that contain nitrogen a reaction rapidly converts it to hydrogen cyanide (HCN) or ammonia (NH_3) and formation of NO takes place by the prompt mechanism [166].

The time available for chemical reactions in gas turbine conditions is small and reactions with slow dissociation rates do not come to completion and thus, concentrations of products are limited to being small relative to their equilibrium values [25]. In such cases the reaction is not reversible. Nonetheless, arrows pointing in both ways are used in the chemical reaction equations presented below even though a particular reaction may not be reversible in gas turbine conditions.

1.2.1 Thermal nitric oxide

At high temperatures nitrogen molecules in the atmospheric air constituent of the reactants oxidise according to the chemical equation 1.1a. The reaction is only relevant at high temperatures because of its high activation energy. The consensus is that at flame temperatures less than $1800K$ the formation of thermal NO is not significant [25, 166]. The reaction forms NO directly

and initiates the reactions in equations 1.1b-c that form more NO.

$$N_2 + O \leftrightharpoons NO + N \tag{1.1a}$$

$$O_2 + N \leftrightharpoons NO + O \tag{1.1b}$$

$$OH + N \leftrightharpoons NO + H \tag{1.1c}$$

The NO formation rate depends exponentially on flame temperature for gaseous and liquid fuels [154]. The NO formation rate with respect to the reactants equivalence ratio (actual to stoichiometric air-fuel ratio) as determined analytically by Correa [25] for typical conditions in laboratory, industrial, and aircraft gas turbines is shown in figure 1.2. The NO formation rate is plotted with respect to the mixture's equivalent ratio (ϕ) which is intrinsically related to the flame temperature. The flame temperature is maximum when the mixture is stoichiometric ($\phi = 1$) and falls exponentially into the fuel-lean ($\phi < 1$) and fuel-rich ($\phi > 1$) sides of stoichiometric. Figure 1.2 clearly shows the analogy between the NO formation rate and the flame temperature. In figure 1.2, it is also important to note that in real gas turbines the NO formation rate can be orders of magnitude higher than in a laboratory setting. Although a low flame temperature can be attained in fuel-lean and fuel-rich combustion the former is preferable because for small combustor residence times there is more than enough oxygen to burn all the fuel and thus, leaving little or no CO and UHC. The low NO_x emissions capability of lean premixed prevaporised (LPM/LPP) and rich-burn quick-quench lean-burn (RQL) combustors stems from operation close to the lean and rich flammability limits at which the flame temperature and consequently the formation rate of thermal NO are low.

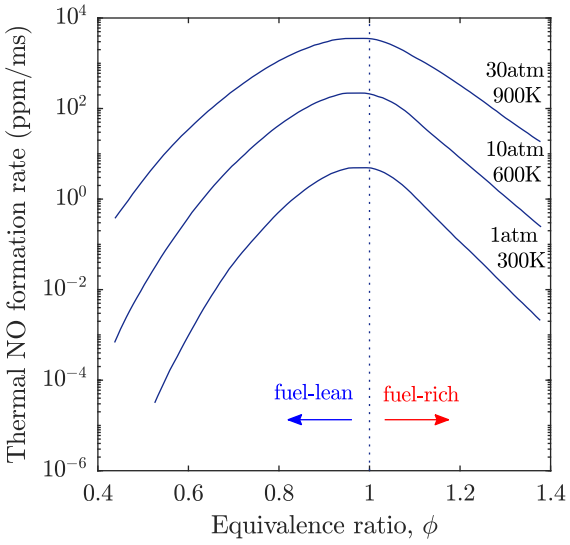

Figure 1.2: Thermal NO formation rate with inlet conditions typical of laboratory gas turbine (1atm, 300K), industrial gas turbine (10atm, 600K) and aero-propulsion gas turbine (30atm, 900K) combustion [25].

1.2.2 Intermediate nitrous oxide

At low flame temperatures nitrogen molecules in the atmospheric air constituent of the reactants oxidise according to the chemical equation 1.2a. The reactants in this reaction are the same as in equation 1.1a but at low temperature conditions the product of the reaction is nitrous oxide (N_2O). The N_2O is an intermediate [166] that then forms NO through the chemical equations 1.2b-d.

$$N_2 + O \leftrightharpoons N_2O \tag{1.2a}$$

$$O + N_2O \leftrightharpoons NO + NO \tag{1.2b}$$

$$H + N_2O \leftrightharpoons NO + NH \tag{1.2c}$$

$$CO + N_2O \leftrightharpoons NO + NCO \tag{1.2d}$$

1.2.3 Prompt nitric oxide

Hydrocarbon radicals react with nitrogen molecules in the atmospheric air constituent of the reactants according to the chemical equation 1.3a. The reaction forms hydrogen cyanide (HCN) and monatomic nitrogen (N). At equivalence ratios $\phi \lesssim 1.2$ the HCN reacts with monatomic oxygen and initiates a chain reaction according to equations 1.3b-e that eventually forms NO. At equivalence ratios $\phi \gtrsim 1.2$ the chemistry becomes complex and is beyond the scope of this thesis which is concerned with the combustion dynamics of fuel-lean mixtures. The interested reader that wishes to know more about the chemical processes can refer to the textbook by Turns [166] and the references therein.

$$CH + N_2 \leftrightharpoons HCN + N \tag{1.3a}$$

$$O + HCN \leftrightharpoons NCO + H \tag{1.3b}$$

$$H + NCO \leftrightharpoons NH + CO \tag{1.3c}$$

$$H + NH \leftrightharpoons N + H_2 \tag{1.3d}$$

$$N + OH \leftrightharpoons NO + H \tag{1.3e}$$

The amount of NO_x formed by the thermal nitric oxide mechanism only becomes comparable to the amounts formed by the other formation mechanisms at low temperatures and in that case the total amount of NO_x is already very small. Hence, flame temperature is a crucial parameter in controlling NO_x emissions.

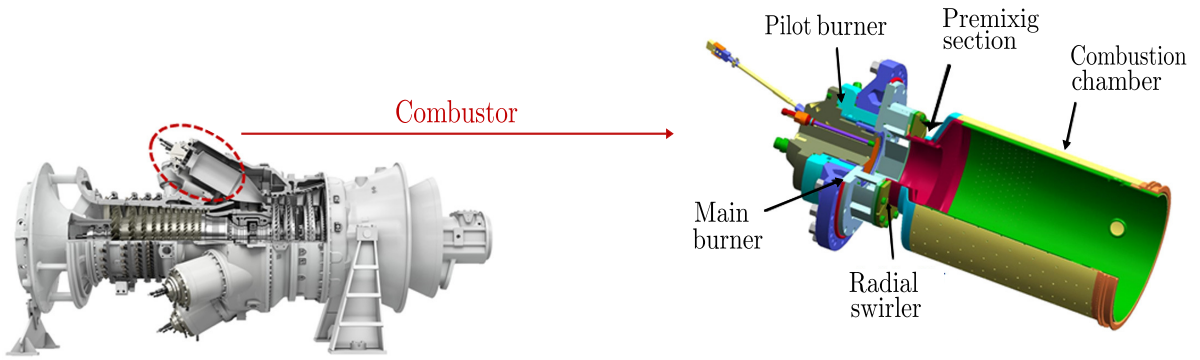

Figure 1.3: Lean premixed combustor used in Siemens industrial gas turbines [106].

1.3 Lean premixed combustion

Lean premixed combustion technology effectively reduces NO_x emissions due to low flame temperature with the additional benefit of enhanced fuel efficiency [35]. The control principle is to supply the flame with a mixture of equivalence ratio that is fuel-lean and spatially homogeneous.

Supplying the flame with a globally fuel-lean mixture is not sufficient for achieving low NO_x emissions. Local regions with equivalence ratio greater than the average value should be eliminated otherwise NO_x emissions will remain high in these regions due to locally high flame temperatures, thus, negating the purpose of lean premixed combustion. Hence, the air and fuel in lean premixed (LPM) combustors are mixed upstream of the flame to ensure a homogeneous mixture of reactants at the flame.

The typical layout of an LPM combustor used in industrial gas turbine engines is shown in figure 1.3 [106]. The component that connects the air-fuel supply system to the combustion chamber in the schematic of figure 1.3 is the premixing section where the fuel and air are premixed. Air is fed to the premixer through a swirler that induces a tangential velocity component to the air stream. The swirl not only enhances the mixing process but also generates a recirculation zone at the dump plane that provides a reduced velocity region where the flame speed and flow velocity can be matched to stabilize the flame [67]. The fuel injection system injects gaseous or atomised/vaporised liquid fuel into the swirling air stream. Atomizers/vaporisers convert the bulk liquid fuel into small droplets or vapor prior to mixing and are vital to ensuring a homogeneous mixture. Lean premixed combustors firing liquid fuels are termed lean premixed prevaporised (LPP) combustors because the atomisation/vaporisation process is an essential component of the technology. The significance of the quality of mixing is exemplified in figure 1.4 that shows NO_x emissions from a General Electric LM6000 gas turbine combustor as a function of average flame temperature for various degrees of premixing [110].

Although, lean premixed combustion is effective in controlling NO_x emissions, combustors utilizing this type of technology are susceptible to a number of dynamical problems including combustion instabilities, flame flashback, and flame blowoff [108].

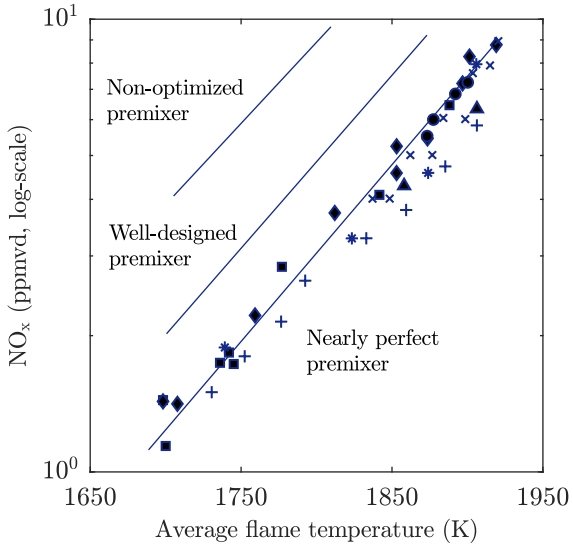

Figure 1.4: Influence of mixing quality on NO_x emissions [110].

1.4 Combustion instabilities

Combustion instabilities are a resonance condition of combustion systems that are characterized by large amplitude oscillations of the fluid and thermodynamic properties of the flow [115]. The large amplitude oscillations can have detrimental effects on system operability and the longevity of its components. Thrust oscillations, severe vibrations that interfere with control systems, enhanced heat transfer and thermal stresses to the combustor walls, fatigue of system components and flame flashback or blowoff are some of the problems caused by combustion instabilities [115]. Damages caused to combustor components due to combustion instabilities and flame flashback are shown in figure 1.5.

In general, a driving mechanism involving many elementary processes generates flow perturbations $(p', q', \dots)$ that become coupled to the driving processes through a feedback mechanism. The perturbations then grow exponentially if damping processes are weak until a limit cycle is reached [115].

Rayleigh [147] recognised that energy is fed into the perturbations when the phase between

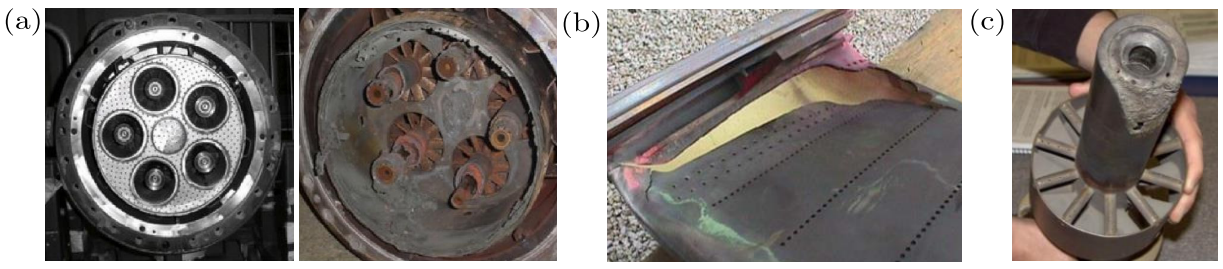

Figure 1.5: Combustor damage due to combustion dynamics related problems [50]: (a) burner nozzles before and after sustaining damage due to combustion instabilities, (b) cracked transition piece, and (c) burner nozzle damage due to flame flashback.

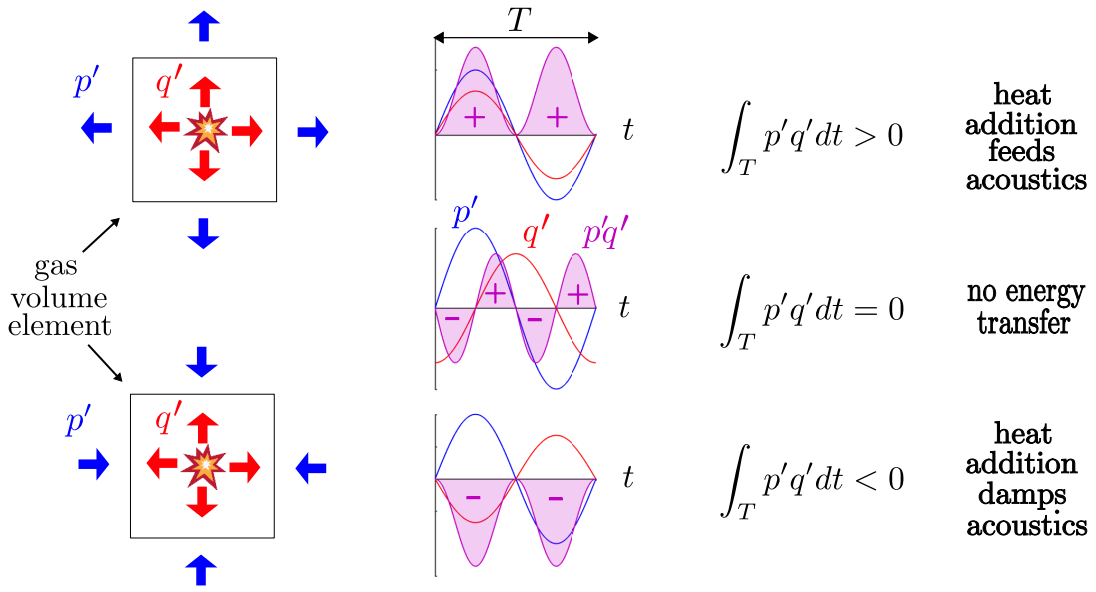

Figure 1.6: Physical interpretation of the Rayleigh criterion.

the fluctuating flow variables and the fluctuating heat release of the flame is less than a quarter cycle ($\pi/2rad$). In physical terms, the condition states that the heat release performs work on or adds energy to the gas when the resulting gas expansion is in phase with its pressure. This is illustrated in figure 1.6. Mathematically the condition is $\int_t p'q'dt > 0$, where p' is the pressure fluctuation, q' is the heat release fluctuation of the flame and t is time.

The Rayleigh criterion [147] determines whether the flow perturbations have the potential to grow. Combustion instability will only occur if the energy being added to the perturbations is not being lost due to energy loss or damping processes D_i, where, $i = 1,\ldots,N$ are all the damping processes in the system [115]. Damping processes include viscous dissipation, heat transfer, and acoustic attenuation at the combustor walls. The condition for combustion instabilities to occur is summarized in the flowchart in figure 1.7. There are numerous elementary processes that can be involved in driving the instability while the feedback mechanism is usually acoustic in nature [18]. Hence, combustion instabilities are also known as thermoacoustic instabilities. It should be noted that this definition of combustion instability excludes the intrinsic flame instabilities [114].

Intrinsic characteristics of combustion systems make them prone to combustion instabilities [18, 20]. Firstly, the power density of these systems is large and even a very small fraction of this energy suffices to drive strong oscillations. This is why combustion instabilities first presented a major problem in high power-density rocket engines. A notable example is the development of the F-1 liquid-propellant rocket engine in the late 1950s that required thousands of costly full-scale tests to eliminate severe combustion instabilities by trial and error [137]. Secondly, damping processes are weak and as long as there is a driving mechanism to feed energy into the flow perturbations they will grow. Lastly, there are time lags associated with

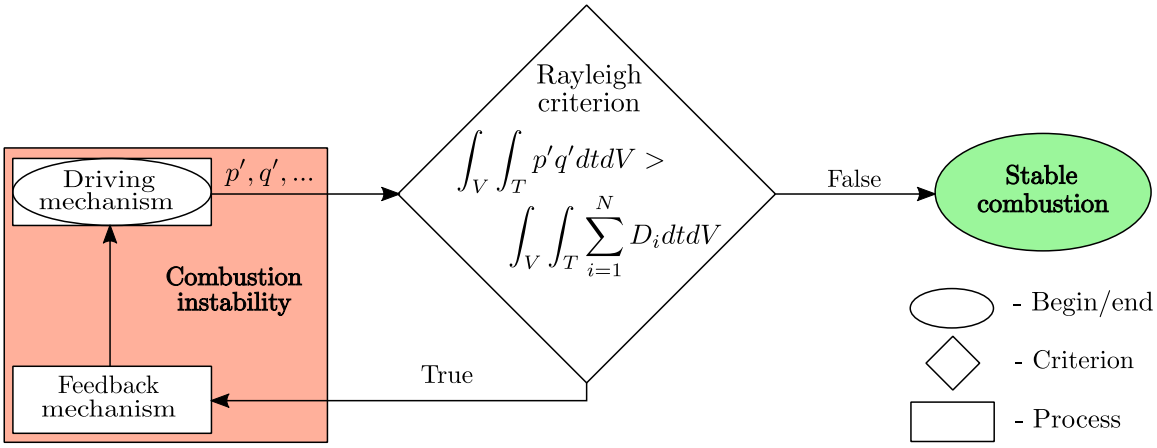

Figure 1.7: Stability of a combustion system.

the numerous elementary processes involved and it is difficult to ensure that none of them will satisfy the Rayleigh criterion. Methods of controlling combustion instabilities may be passive or active. Active control methods are still in their infancy and only combustors that are stable by design (passive control) are suitable for practical applications [143]. The typical disadvantages of passive control are that it is only implementable to a designed combustor over certain narrow frequency ranges and that it does not respond to changes in operating conditions [1]. Active control is more adaptive. However, before active control can become routine in practical applications progress needs to be made in sensors, control algorithms, and actuators. Control algorithms require better understanding of the processes driving the instabilities and their interactions. Sensors and actuators need to be able to operate in the harsh environment of the combustion chamber [18, 20, 26, 143].

The basic interactions that lead to combustion instabilities are illustrated in a block diagram in figure 1.8. Studies of the individual processes concentrate on different stages of the physics. Flame stabilization includes studies of flame flashback and flame blowoff. The block on entropy waves includes studies of entropy wave (1) generation at the flame, (2) propagation through the combustor, (3) acceleration through either nozzles or blade rows to produce acoustic waves known as entropy noise, (4) transmission through downstream components and (5) reflection back into the combustor [131].

Flame flashback and the propagation of entropy waves through the combustor are the focus of the work in this thesis. Results obtained in studies of the elementary processes may be used to build a comprehensive model of the system. However, interactions between the elementary processes still need to be established [18, 21]. Studies of system dynamics circumvent the need of establishing interactions between the elementary processes but at the expense of less insight into the physics. Such studies can be made using tools from dynamical systems theory [83, 158].

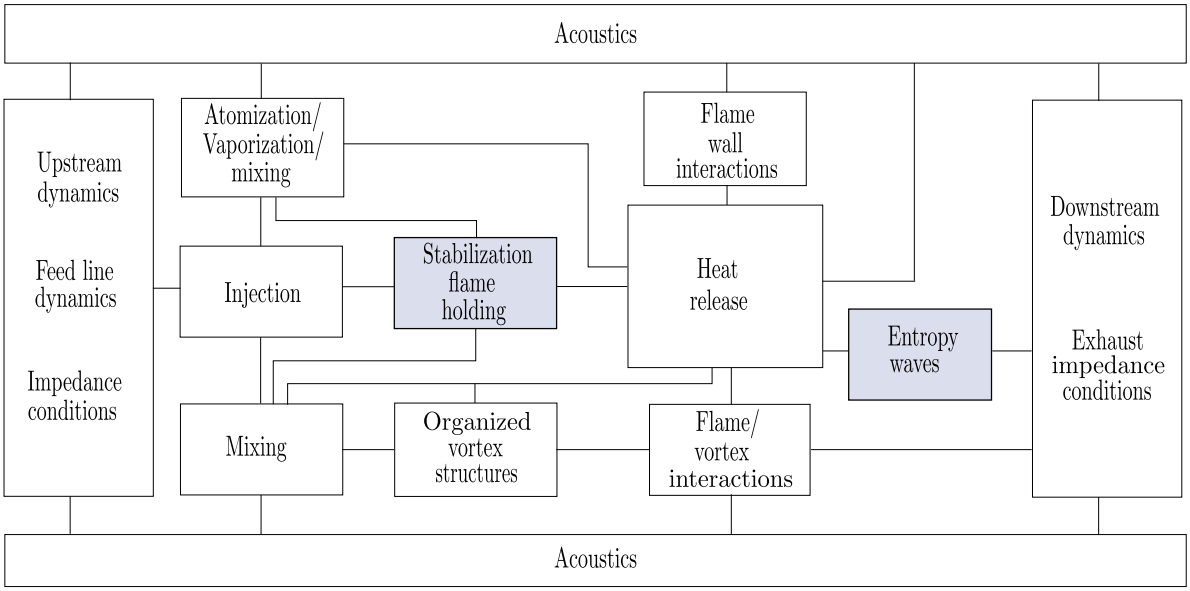

Figure 1.8: Basic interactions leading to combustion instabilities [18].

1.5 Flame flashback

Flame flashback is a transient phenomenon characterized by the sudden propagation of the flame upstream from its designed stabilization point [12]. As such it can only occur in premixed combustion systems where there is a flammable mixture available upstream in the premixing section of the combustor.

Igniting a quiescent mixture of unburned reactants generates a reactive region (the flame) that propagates towards the unburned mixture at a speed known as the laminar flame speed $\overrightarrow{S_l}$ and it is a thermochemical property of the reactive mixture [166]. Supplying the flame with a flowing mixture of unburned reactants at a rate $\overrightarrow{u}$ equal to the flame speed results in a stable (stationary) flame. At a flow speed less than the flame speed the flame propagates upstream at a relative velocity of $\overrightarrow{v_f} = \overrightarrow{S_l} - \overrightarrow{u}$. Contrariwise, the flame propagates downstream. The former case is known as flame flashback and the latter case as flame blowoff. Flames of fuels containing hydrogen are particularly problematic to stabilize due to the higher flame speed of hydrogen compared to those of conventional fuels [165, 29, 11, 159, 160]. A turbulent flow of reactants enhances the flame speed due to the turbulent transport of chemical species and heat. Therefore, the flame speed ($\overrightarrow{S_f}$) is classified as either laminar ($\overrightarrow{S_l}$) or turbulent ($\overrightarrow{S_t}$) depending on the type of flow.

Periodic flame flashback can be triggered by unexpected velocity fluctuations associated with combustion instabilities [91, 163]. The flame will propagate upstream during the part of the oscillation when the flow velocity is less than the flame velocity provided that the oscillations are large enough. In particular, low frequency velocity fluctuations trigger the event because high frequency fluctuations simply do not give the flame enough time to propagate upstream [164, 165]. Subsequently, the flame motion causes a periodic heat release at the flashback frequency

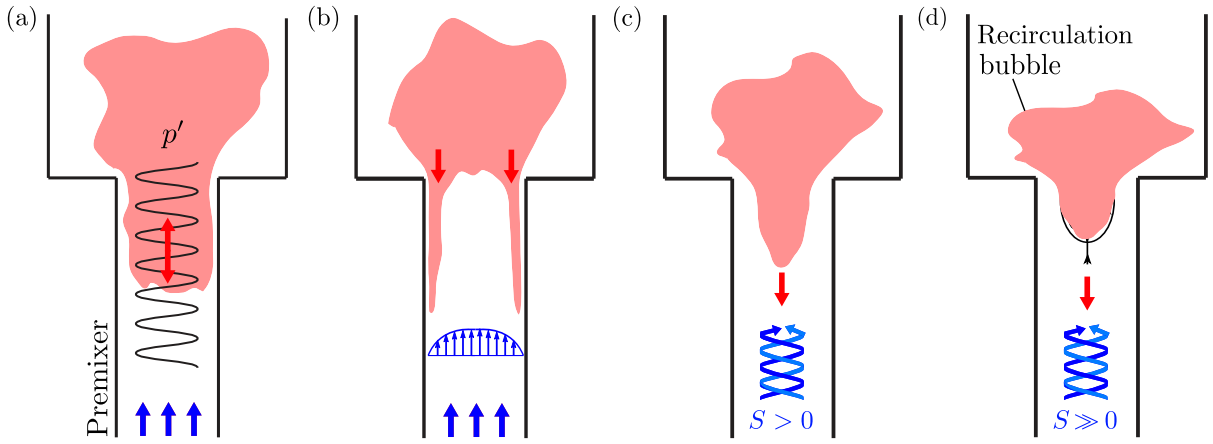

Figure 1.9: Mechanisms of flame flashback: (a) flashback due to combustion instabilities, (b) boundary layer flashback, (c) core flow flashback where $S > 0$ is the swirl number, and (d) CIVB flashback.

that can become coupled with an acoustic mode of the combustor and thus, become a driving mechanism of combustion instability [164, 165].

Even if flame flashback does not become a driving mechanism of combustion instability, the event itself can have destructive effects as combustor components upstream of the designed flame stabilization point are not designed to withstand flame temperatures [108]. The components upstream are therefore, in the event of flashback, at risk of sustaining serious damage or reduced lifespan and any debris carried with the flow through the turbine can cause more damage downstream. The phenomenon can also be triggered in the absence of combustion instabilities. Advanced numerical and experimental studies [55, 102, 155, 56, 100, 95] have identified an additional three flashback mechanisms. These are referred to as flashback in the boundary layer, flashback in the core of a swirling flow, and flashback due to combustion induced vortex breakdown (CIVB). These mechanisms are illustrated in figure 1.9.

In the boundary layers, the flow velocity and the flame speed vary substantially. The freestream velocity is gradually reduced to zero at the wall due to the wall friction. On the other hand, the flame speed is influenced by flame quenching due to the diffusion of heat and chemical species towards the wall and flame stretching due to the velocity gradient [144, 65, 12]. There is therefore potential for the flow velocity and flame speed to become imbalanced in the boundary layer and hence, result in flame flashback. The early work of Lewis and von Elbe [111] introduced the concept of a critical velocity gradient g_c in equation 1.4, where u is the flow velocity, τ_w is the flow shear stress on the wall, μ is the kinematic viscosity, $S_f|_{y=\delta_b}$ is the near wall flame speed, and δ_b is the penetration distance. The penetration distance is the distance from the wall at which the wall effects cease to influence the flame speed [111, 175]. The near wall flame speed and the penetration distance are both difficult to predict and hence, the critical velocity gradient has been determined empirically for various fuel mixtures [29, 49, 152]. The theoretical and experimental studies of laminar reactive boundary layers conducted by Kurdyumov et al. [104, 105]

advanced the findings of Lewis and von Elbe [111]. In these works, the critical condition for the flashback of a flame in a laminar boundary layer is determined for asymptotic limits and single step chemistry.

$$g_c = \left.\frac{\partial u}{\partial y}\right|_{y=0} = \frac{\tau_w}{\mu} = \frac{S_f|_{y=\delta_b}}{\delta_b} \tag{1.4}$$

The case of a turbulent reactive boundary layer in an adverse pressure gradient is studied by Eichler and Sattelmayer [47] using both experimental and numerical approaches. The critical velocity gradients determined in their work are not in good agreement with the predictions made by the theory of Lewis and von Elbe [111]. This is believed to be due to the influence of the adverse pressure gradient on the boundary layer [47]. Later, Eichler and Sattelmayer [48] investigated the flashback of methane and hydrogen flames in both laminar and turbulent boundary layers using micro-PIV. This study reported upstream propagating fronts with large curvature and complex dynamics. Further, regions of back-flow and static pressure gain are observed in front of the flame during the flashback [48]. In a study that followed thereafter, Eichler et al. [49] investigated the flashback of an open flame for different initial conditions. In one case, the flame is initially stable at a location completely out of the upstream duct and in the other case it is slightly inside the duct. The critical velocity gradients from these cases are substantially different [49]. A number of reasons are reported to explain the distinction including differences in the pressure fields near the reactive region and details of the duct configuration [49]. These works [47, 49, 48] clearly demonstrate the significant influence of the pressure field on flame flashback.

A recent direct numerical simulation by Gruber et al. [66] of flame flashback in a turbulent boundary layer included detailed chemistry and showed that the leading edge of the propagating reactive front is always very close to the wall and it is clearly corrugated. Furthermore, the result shows that there are back-flow pockets and small variations in the pressure of the cold flow in the vicinity of the corrugated front [66]. These flow features are argued to be the result of a hydrodynamic instability of the Darrieus-Landau type [66]. On the basis of these findings, Gruber et al. [66] consider the criterion of Lewis and von Elbe [111] to be inadequate for the evaluation of flashback in turbulent boundary layers.

Flashback in the boundary layer is usually not regarded to be significant in swirling flows [56]. In the core of swirling flows, flame flashback can occur due to the interaction of combustion with the swirl aerodynamics, even when the flow velocity is substantially greater than the flame velocity [56, 103, 136]. For this reason, flame propagation in vortices has been studied to provide insight into the interactions between combustion and simple vortical flows [76]. The mechanisms that have been proposed to explain the enhanced flame propagation speed in vortices include, flame kernel deformation [126, 68], vortex bursting [24], baroclinic push [5], and azimuthal vorticity evolution [167]. Flame kernel deformation is the result of the rearrangement of the density field due to centrifugal forces [68]. The higher density unburned mixture moves

away from the axis of rotation and the lower density burned mixture towards it. This rearrangement of the flow results in the flame being 'pushed' on to the axis of rotation and spreading along it [126] in a motion that has been likened to that of a flame spreading beneath a plane ceiling [118]. The vortex bursting theory explains that the enhanced flame propagation speed is due to a pressure rise across the flame that acts to push the flame upstream [24]. The pressure rise is the result of the swirl velocity decreasing across the flame. The treatment of the original theory assumed that the process is similar to a hydraulic jump and equated the pressure forces induced by the rotation of the fluid with the momentum flux due to the flame being 'pulled' inside the vortex [24]. Later, the more detailed approach of Daneshyar and Hill [30] applied the principle of conservation of angular momentum to the vortex tube that is expanding radially across the flame. The conservation of angular momentum across the flame gives the pressure rise in equation 1.5, where, ΔP is the pressure difference across the flame, ρ is density with the subscripts u and b referring to the unburned and burned mixtures respectively, and $V_{\theta_{max}}$ is the maximum tangential velocity. The latest model of the vortex bursting mechanism is the back-pressure drive flame propagation mechanism developed by Ishizuka [76]. The particular model has been found to be in good agreement with experiment [76, 77]. The model applies the principles of conservation of mass, linear momentum, and angular momentum to the vortex tube that is expanding radially across the flame. The baroclinic push and azimuthal vorticity evolution mechanisms both explain that the flame propagation is enhanced by azimuthal vorticity that induces a propagation velocity towards the unburned mixture. The baroclinic push mechanism explains that baroclinic torque generates azimuthal vorticity due to the density and pressure fields being misaligned [5]. On the other hand, the azimuthal vorticity evolution mechanism explains that twisted vortex filaments give rise to azimuthal vorticity as they are twisted by the expanding vortex tube across the flame [167].

$$\Delta P = \rho_u V_{\theta_{max}}^2 \left[1 - \left(\frac{\rho_b}{\rho_u} \right)^2 \right] \tag{1.5}$$

The swirl induced to the flow in many gas turbine combustors is deliberately strong enough to invoke a hydrodynamic instability at the dump plane (premixer exit) known as vortex breakdown [119]. The vortex breakdown generates an internal recirculation zone (IRZ), also known as a recirculation bubble, along the central axis of the swirling flow that is utilized to stabilize the flame [119]. Although this aerodynamic stabilization of the flame is widespread, the phenomenon has been the subject of extensive investigations as it is not yet fully understood [119, 51, 109]. The studies that have been conducted show that positive pressure gradients and sudden flow expansions trigger the vortex breakdown. Most swirling flames poses these features that promote its occurrence. Therefore, the mechanism has also been studied extensively in the case of reactive flows [56, 102, 100, 95, 103, 98]. Fritz et al. [56] studied theoretically and experimentally the flashback of a flame that is stabilized by vortex breakdown. Flashback occurred after the breakdown of the swirling flow field and the flame propagated upstream of

the recirculation bubble. However, Fritz et al. [56] showed that the occurrence of flashback strongly depends upon the combustor configuration and small design changes can prevent it. Nonetheless, the vortex breakdown occurred further upstream in the reactive flow compared to the isothermal flow. Hence, Fritz et al. [56] reported that combustion can enhance the occurrence of vortex breakdown and postulated that the enhancement is due to the pressure changes in the upstream flow caused by the flame heat release. The study therefore concluded that the flashback occurred due to the mechanism of Combustion Induced Vortex Breakdown (CIVB). Kiesewetter et al. [95] conducted a two-dimensional URANS simulation of a swirling flame. The simulation predicted the stability limits determined in the experiment of Fritz et al. [56]. The validity of the simulation is also confirmed later by the results of the experiment of Konle et al. [100] that used high-speed PIV and LIF techniques. The simulation of Kiesewetter et al. [95] solved the vorticity transport equation to subsequently evaluate the influence of each source term in the equation on the upstream motion of the flame. The analysis showed that the baroclinic torque is the most significant contributor to the generation of negative axial velocity. The study argued that the stability of the flame depends upon a flame quenching process that occurs upstream of the flame [95]. Kröner et al. [103] investigated the quenching process that occurs ahead of the flame by modelling the turbulence and chemistry interactions in the case of CIVB. The model of Kröner et al. [103] treated the recirculation bubble ahead of the flame as a perfectly stirred reactor. A burner specific time constant is evaluated experimentally and only needs to be determined once for a specific burner configuration. In the study, the model predicts the stability limits of the burner in various operating conditions but with systematic errors. Konle and Sattelmayer [98] improved the model of Kröner et al. [103] by releasing the assumption of a perfectly stirred reactor for the low turbulent Reynolds numbers. This resulted in a better agreement of the model with the experiment [98]. Experimental studies carried out by Blesinger et al. [14] on swirling flames showed that for high swirl flows the vortex breakdown occurs far upstream of the mixing tube exit and the recirculation zone extends into the mixing tube. In this case, the flashback is then caused by turbulent burning in the axial recirculation zone that extends upstream along the vortex axis into the mixing tube [14]. Flashback in this situation depends on turbulent quenching of the flame and hence, the mechanism is discriminated from CIVB flashback and termed flashback due to Turbulent Burning on the Vortex Axis (TBVA).

Another method of flame stabilization involves using a bluff-body to generate the recirculation region that anchors the flame. Therefore, flame flashback has been also investigated in swirling flows inside a duct with a central bluff-body [114, 135, 69, 153]. Nauert et al. [135] showed that there is a critical swirl number that when exceeded can result in flame flashback [135]. The experiment of Heeger et al. [69] investigated the flashback of a turbulent premixed flame in a swirl burner with a central bluff-body. In the study, the velocity field of the unburned gas is obtained using PIV, the flame front is detected using OH-PLIF and FL (Flame Luminosity) imaging [69]. In a very few of the flashback realisations the study found that there is a negative

velocity region ahead of the flame which drives the flame upstream [69]. Measurements of the flow axial velocity ahead of the flame tip revealed that the flame is propagating in a separated or thickened boundary layer [69]. Heeger et al. [69] postulated that the separation is due to the development of an adverse pressure gradient along the bluff-body. The adverse pressure gradient follows from a decrease in the density due to combustion and a decrease in the circumferential velocity due to the intense mixing with the surrounding air. De and Acharya [33, 34] conducted large eddy simulations (LES) on the flashback of hydrogen enriched premixed flames. The geometry of the computational model is similar to that of the burner used in the experiment of Heeger et al. [69]. The results of the LES simulations of De and Acharya [33, 34] confirmed the negative velocity region ahead of the upstream propagating front observed by Heeger et al. [69]. De and Acharya [33, 34] argued that a positive pressure difference is responsible for the formation of a recirculation bubble ahead of the flame that is propagating upstream. The positive pressure difference is found to be the combined effect of centrifugal forces and density jumps across the flame [33]. Hence, the simulation of De and Acharya [33] confirmed the earlier postulate of Heeger et al. [69]. The existence of a positive pressure difference is the driving force for an upstream propagating front according to the back-pressure mechanism of Ishizuka [76]. In the recent study of Karimi et al. [88], the back-pressure drive mechanism of Ishizuka [76] is extended to a configuration similar to that used in the experiment of Heeger et al. [69]. The pressure difference across the flame that is calculated using the extended back-pressure model is similar to experimentally measured values [88]. The study of Karimi et al. [88] therefore argues that this adverse pressure gradient could contribute to the observed flame flashback [69] by causing the separation of the boundary layer on the surface of the bluff-body. It is important to note that the recent experimental and theoretical investigations of flame flashback in boundary layers also highlight the significance of an adverse pressure gradient on the flashback [47, 49, 48, 114]. Pressure changes can influence the baroclinic torque, which is important in the case of the CIVB mechanism [95, 33, 34]. Thus, the detection of all mechanisms that generate adverse pressure gradients is central to the understanding of flame flashback in configurations involving swirling flows and a central bluff-body.

Regardless of the responsible mechanism, flame flashback is always a highly transient phenomenon, which takes a small fraction of a second for completion. This transient nature of flashback makes it exceedingly difficult to control and suppress. As a result, an early detection of the possibility of flashback is of high significance.

1.6 Entropy waves

An unsteady heat release, at the flame in the case of combustion, generates non-isentropic temperature perturbations known as entropy waves [157]. Entropy waves are generated in addition to isentropic pressure perturbations known as acoustic waves. Unlike acoustic waves that travel

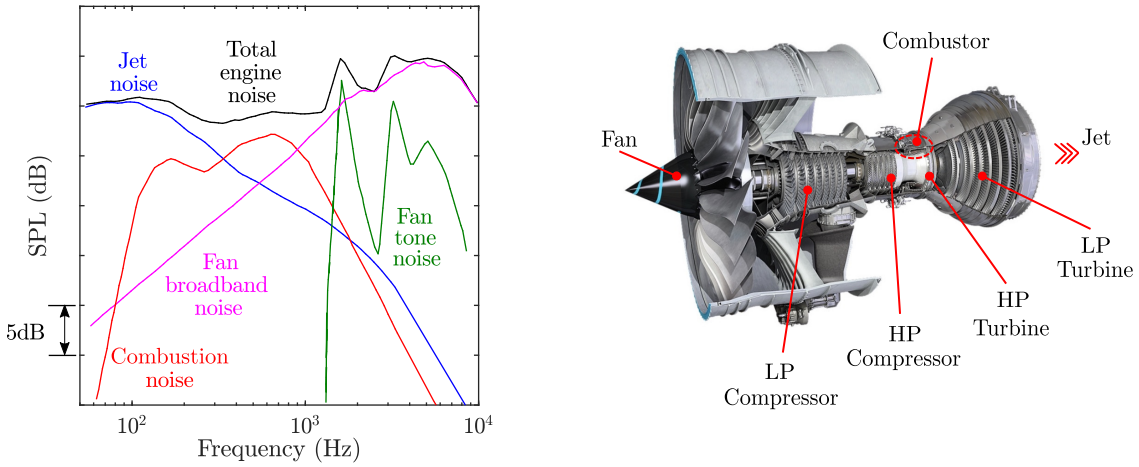

Figure 1.10: Noise from an aircraft gas turbine [39].

at the local speed of sound, entropy waves travel at the local speed of the flow or with the flow. These temperature perturbations (hot and cold spots) are attenuated as they advect through the combustion chamber [151, 132, 58]. The entropy waves that make it through the combustor flow field enter regions with non-zero mean velocity gradient and cause a modulation of the associated pressure gradient generating acoustic waves known as entropy noise [73, 174, 131]. Since entropy noise is not generated at the flame but is a consequence of the unsteady heat release of the flame, it is termed indirect combustion noise whereas noise generated at the flame is termed direct combustion noise.

Entropy perturbations are not the only source of indirect combustion noise. Other non-acoustic perturbations from the flame generate indirect noise on entering regions of non-zero velocity gradient. Indirect noise from vorticity perturbations is known as vorticity noise [27, 97, 39] and that from perturbations in the composition of the burned gas is known as compositional noise [75, 121, 149].

The theoretical study of Leyko et al. [112] has shown that the ratio of indirect to direct combustion noise depends upon the Mach numbers in the flame region and nozzle. On the basis of this Mach number dependence, Leyko et al. [112] argue that indirect noise should become appreciable in real gas turbines. A recent review of combustion noise published the graph in figure 1.10 that shows the typical noise contribution of the turbomachinery in an aircraft turbojet at approach [39]. The graph in figure 1.10 confirms the argument of Leyko et al. [112] that combustion noise in real gas turbines is appreciable and in fact over a broad range of frequencies.

Direct acoustic noise generated at the flame is resolved into reflected and transmitted components at the combustor exit nozzle. Similarly, entropy waves upon reaching the combustor exit nozzle generate reflected and transmitted entropy noise components. These combustion noise components are summarized schematically in figure 1.11. The transmitted component of entropy noise propagates through the turbine row stages and eventually contributes to engine exhaust noise in the case of aircraft gas turbines [131]. The reflected component of the en-

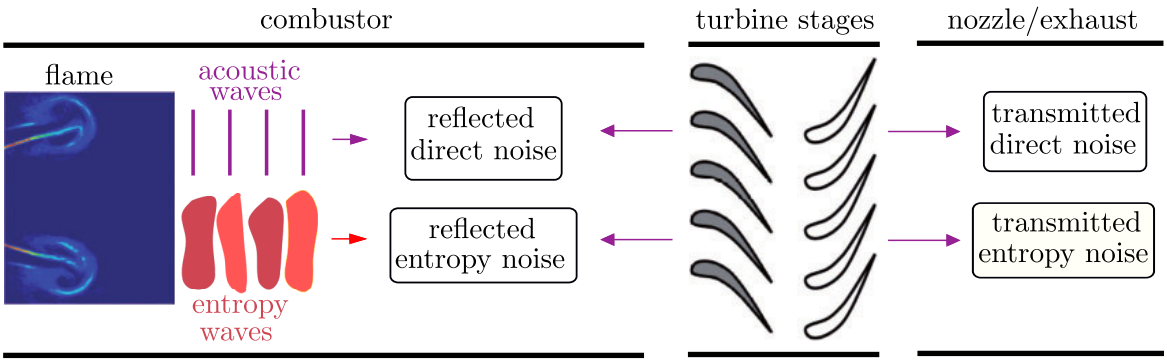

Figure 1.11: The direct acoustic and indirect entropy noise components of combustion noise in gas turbines [131].

tropy noise is more troublesome as it returns to the flame and becomes part of the combustion instability feedback loop [90].

Determining the influence of entropy noise on combustion instability requires knowing the entropy noise levels. However, experimentally obtained measurements of entropy noise are difficult because indirect combustion noise cannot be distinguished from the direct combustion noise in acoustic measurements [36]. Thus, entropy noise levels have to be inferred from measurements of temperature fluctuations at the inlet of the combustor exit nozzle before their passage through the nozzle to generate noise. The earliest experimental attempts on indirect combustion noise are those of Bohn [15] and Zukoski and Auerbach [178]. More recent experimental investigations are those of Hield and Brear [70], Carolan [22], Brear et al. [17], and Rausch et al. [146]. The measurement of high frequency temperature fluctuations is especially challenging due to the slow response of thermocouples and the strongly contaminated signals of laser vibrometers [131]. These and other difficulties associated with conduction of temperature measurements in combustors led to the development of the Entropy Wave Generator (EWG) [7, 8, 9]. This is an experimental set-up for artificially generating temperature disturbances with a heater module that then advect with the bulk flow in a controlled manner. Most recently, Domenico et al. [36] demonstrated for the first time that Laser-Induced Thermal Grating Spectroscopy (LITGS) can be used to measure unsteady temperature and density variations. However, the LITGS technique requires seeding the flow with a species that absorbs the specific wavelength of the laser and this alters the mean flow properties.

Experimentally or numerically obtained temperature fluctuations need to be converted to acoustic fluctuations in order to evaluate the effect of entropy waves on combustion instabilities. Thus, many studies have concentrated on the acoustic response of nozzles to entropic forcing [130, 59, 42, 74]. These studies are mostly based on the seminal work of Marble and Candel [122]. The analysis of Marble and Candel [122] applied the linearised one-dimensional Euler equations to a compact nozzle and derived expressions for the reflected and transmitted acoustic waves that are generated by the passage of entropic and acoustic waves through the nozzle.

Cumpsty and Marble [28] extended the analysis of Marble and Candel [122] to a turbine stage that they modelled as a quasi-two-dimensional domain. Their study found that the entropy noise is strongly affected by the pressure ratio in each stage [28]. In a later study, Cumpsty [27] compares pressure, entropy, and vorticity noise in a theoretical analysis of unsteady combustion and shows that entropy noise is dominant. These models have recently been extended to less restrictive conditions and examined more thoroughly. Stow et al. [156] remove the assumption of a compact nozzle and using a frequency asymptotic expansion for the flow perturbations calculate the reflection characteristics of an annular nozzle. The approach of Stow et al. [156] is similar to that taken later by Goh and Morgans [59] in their quasi one-dimensional analysis of a non-compact nozzle. Goh and Morgans [59] derive theoretical expressions for the phase of the transmission coefficients due to an incident acoustic or entropic wave and compare them with numerical simulations. Moase et al. [130] uses hyper-geometric functions in a quasi-one-dimensional, analytical investigation of the dynamic response of a nozzle with an arbitrary shape subject to acoustic and entropic excitation. The study of Moase et al. [130] also investigates the effects of non-linearities and develops a method to quantify the degree of nonlinearity. The experiments of Bake et al. [7, 8] and Bake et al. [9] are the first experiments to use the entropy wave generator. These experiments have made available comprehensive point measurements of temperature and acoustic pressure (corrected for external noise) that were recorded as entropy waves passed through super-critical and sub-critical exit nozzles to produce entropy noise. Bake et al. [8] show that the measurements are in qualitative agreement with the analytical work of Marble and Candel [122]. Further, the studies found that the Mach number of the flow significantly influences the conversion of the entropy waves to acoustic waves [7, 8, 9]. Leyko et al. [113] simulate the experiment of Bake et al. [9] using large eddy simulation and also show that the compact analysis of Marble and Candel [122] is able to satisfactorily predict the experimental results. The analytical and numerical investigation of Duran et al. [42, 43] considers the subsonic case of the experiment of Bake et al. [9]. This study reveals through a series of case studies that the compact analysis of Marble and Candel [122] is only able to predict the noise generation at low Mach numbers [42, 43]. Hence, in a later study, Duran and Moreau [41] use the Magnus expansion to release the assumption of a compact nozzle. The study derives analytical expressions for the dynamic response of the transmitted noise and shows that this is highly frequency dependent [41]. Huet and Giauque [74] keeping the assumption of a compact nozzle extends the analysis of Marble and Candel [122] to include high amplitude acoustic and entropic waves. Thus, Huet and Giauque [74] develop a nonlinear compact model for the prediction of the nozzle response to large amplitude forcing. Recently, Ihme [75] identified compositional noise as an additional indirect noise contribution by extending the theory of Marble and Candel [122] for a subcritical nozzle to a multi-component gas mixture. Magri et al. [121] used the extended theory of Ihme [75] to derive transfer functions for the same compact-nozzle conditions considered by Marble and Candel [122] and compared the relative contributions of direct,

entropy, and compositional noise. Their results showed that compositional noise can become comparable to and even exceed entropy noise for fuel-lean conditions. In light of their analytical findings, they highlighted the need for experimental measurements to evaluate the level of compositional noise at the combustor exit of gas turbine engines [121]. Rolland et al. [149] have made experimental measurements of the direct, entropy, and compositional noise resulting from the injection and convection of helium disturbances through a chocked nozzle. Their study compares the experimental results to those from analytical models and their measurements of entropy and compositional noise are in good agreement with the extended Marble and Candel [122] theory of Magri et al. [121].

The influence of the entropy waves on combustion instability is a subject of ongoing research [145, 46, 61, 131]. The experimental work of Macquisten and Dowling [120] and the theoretical work of Dowling [38] showed that the boundary condition at the downstream end of a combustor is of significance to thermo-acoustic stability analysis. A simple model developed by Keller [90] for an unstable combustor shows that the main driver of instabilities are the acoustic waves generated by the advection of entropy disturbances through the chocked exhaust nozzle. The study of Zhu et al. [176] also considers the entropy wave mechanism to be an important driver of the low-frequency combustion instability in their numerical simulation of an unstable spray combustor. A linear model developed by Polifke et al. [145] for an unstable combustor with a chocked exit shows that the interactions between the generated entropy waves and the chocked exit nozzle could alter the thermoacoustic stability of the system. This is confirmed by the experiments of Hield et al. [71] on a thermoacoustically unstable, premixed combustor with open and choked exit nozzles. Furthermore, the work of Hield et al. [71] shows that a model that includes dispersive entropy waves and the boundary conditions of Marble and Candel [122] captures the thermoacoustic instability from the experiment. Hence, Hield et al. [71] conclude that entropy waves are of significance in thermoacoustic stability of combustors [71]. The experimental and theoretical work of Eckstein et al. [46], Eckstein and Sattelmayer [45] on a liquid fuel RQL combustor make the opposite argument. That is, the influence of the entropy wave mechanism on the thermoacoustic instability of the combustor is negligible. The model developed in the theoretical work of Eckstein and Sattelmayer [45] includes the dispersion model developed by Sattelmayer [151]. The early dispersion model of Sattelmayer [151] is the first to somewhat account for the influence of combustion chamber aerodynamics on an advecting entropy perturbation. The modelling approach of Sattelmayer [151] treats an experimental dual fuel burner as a single-input single-output (SISO) dynamical system. The impulse response of the system is taken as the probability density function (PDF) of the residence time and is modelled by a rectangular pulse to yield an analytic expression for the system transfer function. The gain of the system as a function of the forcing frequency leads Sattelmayer [151] to argue that entropy waves are of little significance in the analysis of thermoacoustic instabilities. Experimental measurements of entropy waves inside thermoacoustically unstable combustors confirm that entropy waves could

be highly dispersive [70, 17]. However, the direct numerical simulation of Morgans et al. [132] in an incompressible, non-reactive, channel flow shows that entropy waves could survive the flow dissipation and dispersion effects. Morgans et al. [132] demonstrates that the system response, which Sattelmayer [151] models with a rectangular distribution, is better modelled by a Gaussian pulse. Goh and Morgans [60] used the Gaussian dispersion model and added a dissipation factor and from their model results concluded that entropy waves could significantly modify the thermoacoustic instability of the system. In particular, the model of Goh and Morgans [60] shows that the thermoacoustic instability could be either encouraged or discouraged depending upon the strength of the dissipation and dispersion of the entropy waves. The large eddy simulation of an aero-engine combustor by Motheau et al. [134] shows that entropy waves shift the eigenmodes of the system to higher frequencies and could cause mixed acoustic-entropic instabilities. Jean-Michel Lourier and Aigner [78] emphasize the effect of the shape of entropy waves on the peak pressure fluctuations and argue that the more uniform entropy waves generate weaker pressure waves. From this finding it is clear that dispersion mechanisms can significantly influence the entropy noise generated at the nozzle. Recently, Giusti et al. [58] carried out a large eddy simulation of a turbulent flow with sinusoidal entropy oscillations and developed a theoretical model to predict the decay of entropy waves. The study of Giusti et al. [58] finds that the entropy transfer function scales well with a Helmholtz number based on the entropy wavelength and the streamwise coordinate. At low Helmholtz numbers the decay of the entropy wave is attributed mainly to shear dispersion arising from the spatially variable mean velocity profile and at high Helmholtz numbers turbulent mixing and diffusion also become important [58].

The advection of entropy waves through the combustor flow field has not been investigated in detail compared to their conversion to acoustic waves. The former stage of the physics is equally important and merits more investigation because if entropy waves do not make it to the combustor exit nozzle to start with, then entropy noise is not an issue. Low order models developed in the many nozzle response studies and limited entropy wave advection studies are one-dimensional in keeping with what Sattelmayer [151] referred to as the thermoacoustic paradigm. That is, thermoacoustic instability is ultimately influenced by acoustic waves whether direct or indirect, which exhibit a one-dimensional character in slender confinements, such as in a gas turbine combustor. Hence, so far, entropy waves have also conveniently been given a one-dimensional representation.

1.7 Research needs

The literature on flame flashback concentrates on understanding the physics of individual flashback mechanisms. Ultimately, the results from these studies will be used to design suppression devices for flame flashback that will prevent its occurrence in practical combustion systems. However, active suppression systems that are designed to suppress individual flashback mech-

anisms will require the placement of sensors at multiple and precise locations throughout the combustor. Placement of multiple sensors in the combustor could have a significant influence on system performance. Furthermore, it may not be practically feasible to place sensors at all the desired locations in the combustor. There is therefore the need to be able to detect in advance an approaching flame flashback regardless of the mechanism using a minimum number of sensors in the combustor. System dynamics can be studied from a single point measurement assuming that the dynamics of the measured quantity are representative of the dynamics of the full system. Since flame flashback by any mechanism is a system response to certain conditions the early detection of the phenomenon through processing of system dynamics merits investigation.

The literature on entropy waves consists primarily of studies on the acoustic response of nozzles to entropic forcing. The attenuation of advecting entropy waves which has received little attention in the literature merits more investigation since entropy noise is only generated in the nozzle if the entropy waves survive their advection through the combustion chamber. Furthermore, the frequency content of the entropy noise depends on the state of the entropy waves at the nozzle inlet which is determined by the advection process. The studies on advecting entropy waves have proposed low order models that could be used to replace timely and costly numerical simulations of the reactive flow field in the combustion chamber. However, the models are for one-dimensional entropy waves despite evidence that entropy waves could become spatially uncorrelated by the time they reach the exit nozzle. Furthermore, they have ignored the influence of heat transfer at the walls despite that entropy waves are temperature inhomogeneities. There is therefore the need for a new model for an advecting two-dimensional entropy wave in heat transferring flow. Existing models are also based on the assumption that the amplitude of the entropy waves is small and so the dynamics of amplitude decay are linear. However, it is not known at which amplitude the linearity assumption begins to break down. The limits of linearity need to be identified.

1.8 Objectives

The first objective of the current work is to identify precursors of flame flashback through analysis of existing pressure time series that have been recorded from flame flashback experiments in laboratory scale swirl burners. The pressure time series will be analysed using nonlinear time series analysis (NTSA) since the governing equations of the combustion system contain nonlinearities such as the convective term in the Navier-Stokes equations of fluid dynamics. The analysis will use more than one method for the detection of precursors in order to reinforce the findings. Furthermore, detected precursors can be checked for universality since the pressure time series that are used in the current work are from different swirl burners with dynamically different flows (thermoacoustically stable and unstable).

The second objective of the current work is to develop a model for an advecting two-

dimensional entropy wave in heat transferring flow that can predict the evolution of both the shape and amplitude of the entropy wave. A direct numerical simulation (DNS) similar to those of existing studies will be performed to obtain the evolution of the shape and the amplitude of an advecting entropy wave. The DNS data will then be reduced using a novel methodology to build the low order model.

1.9 Overview

The NTSA methods used in the work on flame flashback and the numerical details of the DNS flow solver used in the work on advecting entropy waves are presented in chapter 2. The studies on flame flashback and advecting entropy waves are in chapters 3 and 4, respectively. A summary of the conclusions of the current work and suggestions for future research are in chapter 5. Appendix A contains the main MATLAB codes written to perform the post-processing of the experimental and numerical datasets. Appendix B contains the code added to the DNS flow solver that is used in the entropy wave advection study.

Chapter 2

Theoretical and numerical methods

2.1 Nonlinear time series analysis (NTSA) methods

2.1.1 Phase space

The temporal behaviour of a dynamical system is described by differential equations that are obtained through mathematical modelling of the system. The time evolution of the state x_i of a dynamical system with $i = 1,\ldots,k$ degrees of freedom is governed by the general system of equations 2.1, where overdots denote derivatives with respect to time ($\dot{x}_i = dx_i/dt$).

$$\begin{aligned} \dot{x_1} &= f_1(x_1,\ldots,x_k) \\ &\vdots \\ \dot{x_k} &= f_k(x_1,\ldots,x_k) \end{aligned} \tag{2.1}$$

The time evolution of the state $x_i(t)$ of the system can be represented geometrically in a space with coordinates x_i. The space is called phase space and a point in this space, called a phase point, represents the state of the system at a specific point in time. The phase point of an initial state $x_i(0)$ traces a curve in phase space as the state of the system changes in time. The curve is called a trajectory and represents the solution of the governing equation 2.1 for initial condition $[x_i(0), \dot{x}_i(0)]$. Phase space is filled with trajectories since any point in phase space can be an initial condition. The phase portrait of a system is a plot in phase space of trajectories showing typical dynamical behaviour of the system.

A mass m hanging from a linear spring of stiffness k that is fixed at the other end is a classical dynamical system. Disturbing the position of the mass from equilibrium sets the system in motion. Mathematical modelling of the system gives that the position of the mass is governed by the linear differential equation 2.2, where $\omega^2 = k/m$.

$$\ddot{x} + \omega^2 x = 0 \tag{2.2}$$

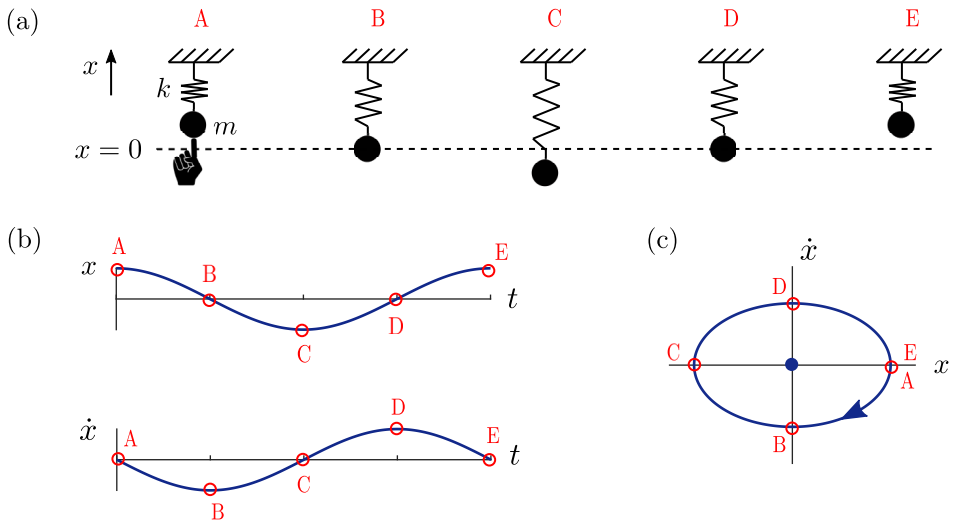

Figure 2.1: Simple harmonic oscillator: (a) dynamical system, (b) time traces of the states, and (c) phase portrait.

The general solution of equation 2.2 is equation 2.3, where C_1 and C_2 are integration constants and are determined from the initial condition. The velocity of the mass is the derivative of position with respect to time and is thus given by equation 2.4.

$$x(t) = C_1 sin(\omega t) + C_2 cos(\omega t) \tag{2.3}$$

$$\dot{x}(t) = \omega C_1 cos(\omega t) - \omega C_2 sin(\omega t) \tag{2.4}$$

Consider the case of a mass that is displaced from its equilibrium position at $x = 0$ to a new position $x(0) = X_0$ and then released with an initial velocity $\dot{x}(0) = 0$. Evaluating equations 2.3 and 2.4 simultaneously for the aforementioned initial condition gives $C_1 = 0$ and $C_2 = X_0$. Hence, the position and velocity of the mass changes in time according to equations 2.5 and 2.6. A complete cycle of the motion is shown in figure 2.1a with the corresponding time traces of position and velocity shown in figure 2.1b.

$$x(t) = X_0 cos(\omega t) \tag{2.5}$$

$$\dot{x}(t) = -\omega X_0 sin(\omega t) \tag{2.6}$$

The governing equation 2.2 of the mass-spring system can be rewritten as equation 2.7, which is in the form of equation 2.1, by introducing the new variables $x_1 = x$ and $x_2 = \dot{x}$. Hence, generating coordinate pairs from equations 2.5 and 2.6 and plotting points at these coordinates in a two-dimensional phase space traces a typical trajectory of the system as the one shown in figure 2.1c. The trajectory circulates around the origin and returns to its starting point defined by the initial condition, thus forming a closed orbit. Hence, closed orbits in phase space correspond to periodic motion. The origin of the coordinate system in phase space corresponds to the state

of $x = 0$ and $\dot{x} = 0$, which is the steady state solution of the governing equation 2.2. A mass with this initial condition would remain at rest forever. A point in phase space that represents a steady state of the system is called a fixed point and correspond to a static equilibrium of the system.

$$\begin{aligned} \dot{x_1} &= x_2 \\ \dot{x_2} &= -\omega^2 x_1 \end{aligned} \tag{2.7}$$

The example above of the simple harmonic oscillator exemplifies that characteristics of a dynamical system such as steady states and periodicity have signature geometric structures in phase space. Hence, phase portraits of dynamical systems can reveal significant information regarding the underlying dynamics of a system. In the case of the simple harmonic oscillator the governing differential equation is linear and can be solved analytically. The analytic functions obtained as such for the states can be used to predict the state of the system at any point in time and for any initial condition. However, systems that are governed by nonlinear differential equations in most cases cannot be solved analytically and for certain values of the parameters in the equations the solution can become extremely sensitive to the initial condition, that is, the solution can be chaotic. In these cases the geometric approach can prove to be very powerful tool for investigating the dynamics of a complex system.

The Lorenz equations 2.8 are a system of nonlinear differential equations that Ed Lorenz derived from a simplified model of convection rolls in the atmosphere [117]. For a range of the parameter r the time-asymptotic state is chaotic with a broadband frequency spectrum. The equations 2.8 with $\sigma = 16$, $b = 4$, and $r = 45.92$ for which r is above the chaotic range are solved here using a forth-order Runge-Kutta integrator for time step $dt = 0.001$. Time series of the states for two initial conditions are shown in figure 2.2a. The two initial conditions result in time series that are qualitatively different although the governing equations are the same. However, the geometry of the trajectories in phase space that are shown in figure 2.2b is qualitatively the same. In fact, the trajectories of the chaotic system always settle onto a complicated set with the same geometric structure. The geometric object is called a strange attractor or the Lorenz attractor in this case.

$$\begin{aligned} \dot{x_1} &= \sigma(x_2 - x_1) \\ \dot{x_2} &= r x_1 - x_2 - x_1 x_3 \\ \dot{x_3} &= x_1 x_2 - b x_3 \end{aligned} \tag{2.8}$$

The dynamics of multi-physics systems that consist of various physical processes are difficult to study through solution of the governing equations because there is not a comprehensive set of equations that include the complex interactions between the various processes [94]. However, there are methods for capturing the structure of a dynamical system in phase space from time

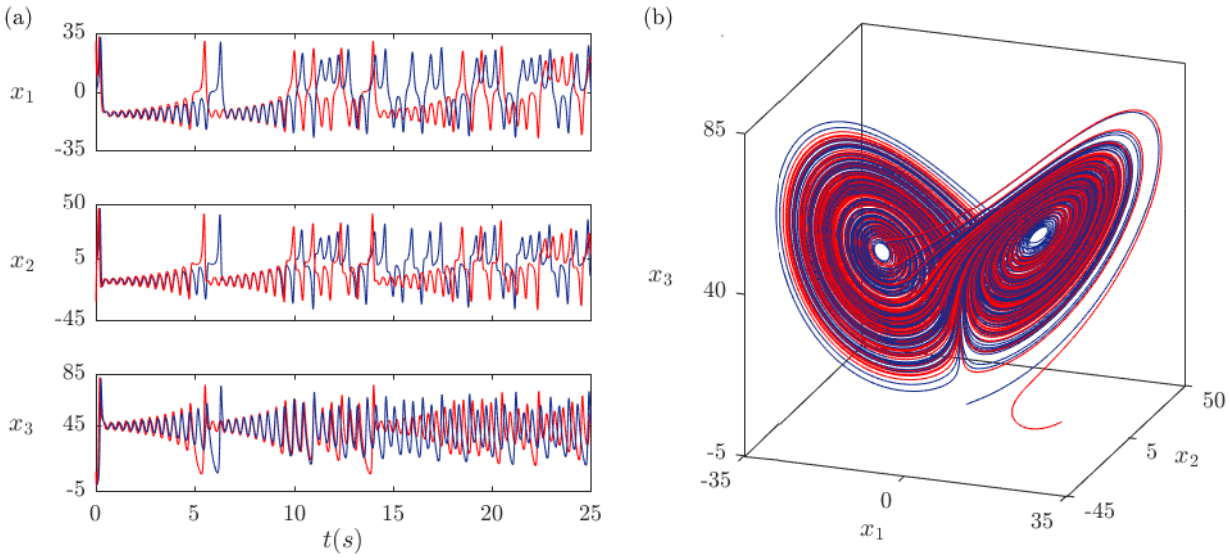

Figure 2.2: Numerical solution of the Lorenz equations ($\sigma = 16, r = 45.92, b = 4$) for two initial conditions using a forth-order Runge-Kutta integrator for time step $dt = 0.001$: (a) time series of the states for the first 25,000 time steps and, (b) phase portrait constructed using 100,000 time steps.

series measurements of at least one of the physical variables [139, 162, 107, 128]. Hence, using this approach, qualitative information about the underlying dynamics can be obtained without having to solve or even derive a comprehensive set of governing equations. The most popular method of phase space reconstruction from time series is that of time-delay embedding [162].

2.1.2 Time-delay embedding

Time-delay embedding generates coordinates of the trajectory for each dimension $d = 1, \ldots, D$ of a D-dimensional phase space by applying to the measured time series a time lag $(d-1)T\tau_s$ that is commensurate with the dimension d and some multiple T of the sampling time τ_s. Hence, the position vector $\overrightarrow{s}(n)$ of point $n = 1, \ldots, N$, where N is the total number of phase points, has the time delayed co-ordinates in equation 2.9.

$$\overrightarrow{s}(n) = [s(n), s(n+T), \ldots, s(n+(D-1)T)] \tag{2.9}$$

The choice of appropriate values for T and D is made on a case-by-case basis using selection criteria [2]. In the current work, the T is chosen to minimize the average mutual information (AMI) between the lagged coordinates [53] and the D is chosen to eliminate the number of false nearest neighbouring phase points in the phase space [92].

Choosing a time delay T that is too small results in lagged coordinates that are nearly identical, whereas choosing a T that is too large results in lagged coordinates that are totally independent. The appropriate choice of the T should result in lagged coordinates that are independent

but not in a statistical sense. Average mutual information is a quantity that is defined in information theory and measures the mutual dependence between two variables [57]. AMI is a more robust measure of mutual dependence than the usual correlation coefficients that are more sensitive to linear dependence. Fraser and Swinney [53] suggest choosing the T that corresponds to the first minimum of the AMI between lagged coordinates. The AMI between lagged coordinates is given by equation 2.10, where N is the total number of phase points (coordinate sets), T is the time delay, $d = 1,\ldots,D$ is a dimension of the D-dimensional phase space, and $P\{\}$ is, depending on the argument, the probability of finding the data point n in the time series $s(n+T(d-1))$ and/or $s(n+Td)$. The plot of the AMI as a function of T for the time series of x_1 from the Lorenz system is shown in figure 2.3a. For the case of the Lorenz system in figure 2.3 the first minimum of the AMI is at $T = 96$.

$$AMI(T) = \sum_{n=1}^{N} P\{s(n+T(d-1)), s(n+Td)\} \log_2 \left[\frac{P\{s(n+T(d-1)), s(n+Td)\}}{P\{s(n+T(d-1))\} P\{s(n+Td)\}} \right] \quad (2.10)$$

Embedding a trajectory in a phase space with insufficient dimensions results in false neighbouring phase points. For example, the trajectory of the simple harmonic oscillator in figure 2.1c is a closed orbit in a two-dimensional phase space. In a one-dimensional phase space, the trajectory is "folded" and appears to be a line. Phase points that are not neighbouring in the two-dimensional space are neighbouring in the one-dimensional space. In a three-dimensional phase space or higher the trajectory remains a closed orbit. Thus, for a system of unknown dimensionality, the appropriate D is the smallest dimension for which the number of false nearest neighbours (FNN) is zero. For an infinite time series, choosing a D that is larger than the minimum value found through the FNN method will not influence the reconstructed phase portrait. However, when reconstructing the trajectory from a finite time series, choosing a larger D than the minimum is not recommended because this results in less coordinate pairs for the embedding, thus shortening the length of the reconstructed trajectory. The plot of the percentage of FNN as a function of D for the time series of x_1 from the Lorenz system is shown in figure 2.3b. In figure 2.3b, the D that brings the number of FNN to zero is $D = 3$, which agrees with the dimensionality of the Lorenz system.

The phase portrait of the Lorenz system is reconstructed from the x_1 time series, which is obtained from numerically solving the Lorenz equations 2.8, using the time-delay embedding method with time delay $T = 96$ and embedding dimension $D = 3$. The phase portrait of the Lorenz system constructed using the numerical solution of the equations 2.8 is shown in figure 2.4a and the reconstructed phase portrait from time-delay embedding of the x_1 time series is shown in figure 2.4b. The reconstructed Lorenz attractor is similar to the original although slightly distorted. The distortion is not surprising since the time-delayed coordinates are not the states of the system. Hence, the geometric structure of a system in phase space can be obtained simply by measuring a signal from the system. In the case of complex combustion phenomena,

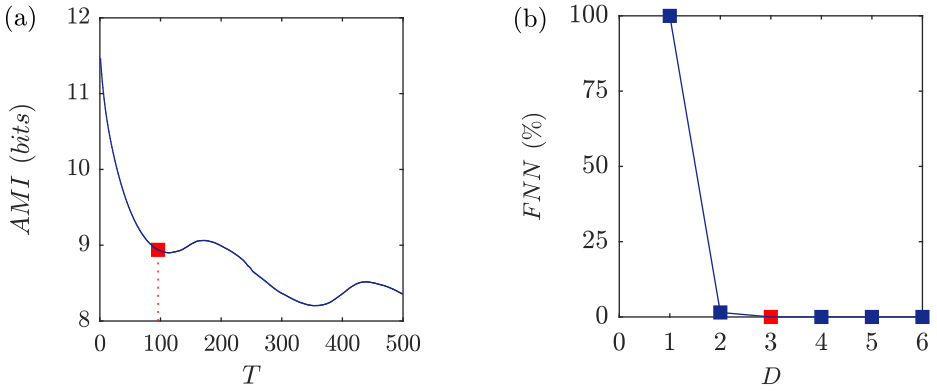

Figure 2.3: Selection criteria for the time delay T and the embedding dimension D for the x_1 time series from the Lorenz system: (a) average mutual information as a function of the time delay T, and (b) percentage of false nearest neighbours as a function of the embedding dimension D.

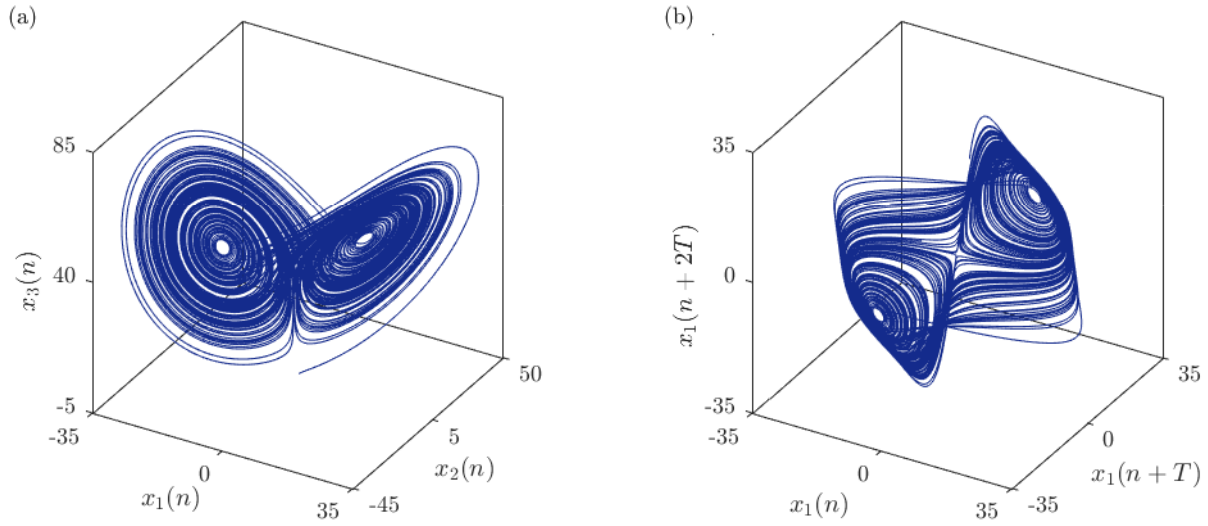

Figure 2.4: Reconstruction of the Lorenz attractor ($\sigma = 16$, $b = 4$, and $r = 45.92$) by embedding time series $x_1(t)$ in a three-dimensional phase space ($T = 96$ and $D = 3$).

the signal could be a pressure time series measured from a pressure transducer on the walls of the combustion chamber.

2.1.3 Translation error of trajectories

The trajectories of a deterministic dynamical system, which is a system that evolves in time according to a governing set of equations, should have an organized structure in phase space. For example, the trajectory of a mass-spring system is a closed orbit in the shape of an ellipse. The trajectory of a mass-spring-damper system spirals toward the fixed point representing the equilibrium position of the mass. The trajectory of the chaotic Lorenz system forms a pattern resembling a a pair of butterfly wings even though the states of the system as a function of time appear to be random. Therefore, a measure of the degree of parallelism of neighbouring sections

of a trajectory gives an indication of the degree of determinism of the system dynamics.

The algorithm of Wayland et al. [172], which is a simpler variant of that of Kaplan and Glass [84], calculates a quantity called the translation error e_{trans} as a measure of the degree of parallelism of neighbouring sections of a trajectory in phase space. The translation error is calculated at various locations along the trajectory. From the N phase points $\overrightarrow{s}(n)$, where $n = 1,\ldots,N$, that make up the trajectory, M are selected at random and are termed random centre points (RCPs). The regions around the RCPs are the localities where the translation error of the trajectories is to be measured. At the RCP a search is done to locate the K nearest neighbouring phase points $\overrightarrow{s}_k(n)$, where $k = 1,\ldots,K$. The tangent of the trajectory at these points is approximated by the translation vector $\overrightarrow{v}_k = \overrightarrow{s}_k(n+1) - \overrightarrow{s}_k(n)$. The translation error in the locality of the RCP is given by equation 2.11, where $\langle \overrightarrow{v}_k \rangle$ is the average translation vector and $||\overrightarrow{v}||$ denotes the Euclidean norm. The average translation vector is given by equation 2.12. The physical interpretation of the average translation vector is shown in figure 2.5 for the case using the $K = 3$ nearest neighbouring points of the RCP. In the current work a $3 \leq K \leq 5$ produced the same results. For neighbouring sections of a trajectory that are parallel, the translation vectors will have similar direction and magnitude, as long as the sampling frequency of the measured time series is large enough for reasons that will be explained below. In this case, the average translation vector will be equal to the individual translation vectors and from equation 2.11 the translation error will be zero. Thus, a zero translation error indicates that the neighbouring sections of the trajectory are perfectly aligned and a non-zero translation error indicates that they are misaligned with the actual value indicating the degree of misalignment.

$$e_{trans} = \frac{1}{K} \sum_{j=1}^{k} \frac{||\overrightarrow{v_k} - \langle \overrightarrow{v_k} \rangle||^2}{||\langle \overrightarrow{v_k} \rangle||^2} \tag{2.11}$$

$$\langle \overrightarrow{v_k} \rangle = \frac{1}{K} \sum_{j=1}^{k} \overrightarrow{v_k} \tag{2.12}$$

It should be noted that the translation error calculation can be influenced by the sampling frequency f_s of the data points in the time series that produced the lagged coordinates for the phase portrait reconstruction. For a low f_s, the phase points are sparsely packed along the trajectory and the translation vectors around the RCPs are not a good approximation of the tangent of the trajectory. Thus, for time series measured at a low sampling rate the translation error may not be a good measure of the degree of parallelism of the neighbouring sections of the trajectory. Contrariwise, for a high f_s, the phase points are densely packed along the trajectory and the translation vectors are a good approximation of the tangent of the trajectory and choosing a sufficient K should give a translation error that is a good measure of the degree of parallelism of the neighbouring sections of the trajectory. Hence, it would be best practice during the measurement campaign to make time series measurements using a few sampling

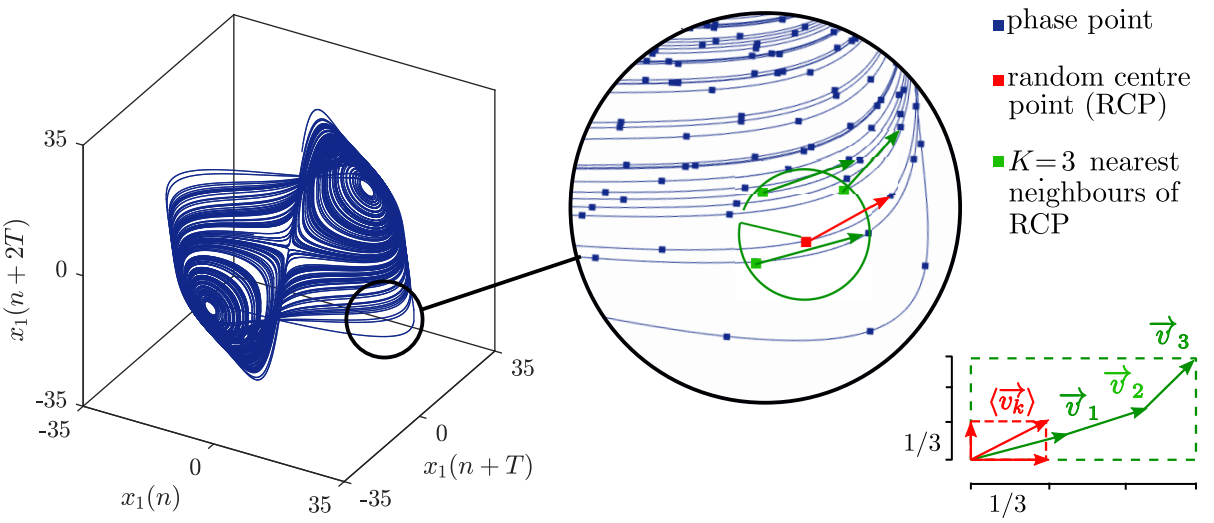

Figure 2.5: Physical interpretation of the average translation vector calculation in equation 2.12. A phase point on the trajectory of the reconstructed Lorenz attractor is taken as a random centre point (RCP). The translation vectors of the $K = 3$ nearest neighbouring phase points are used to determine the average translation vector for the region around the RCP.

frequencies so as to check that the calculated translation error is independent of the sampling frequency.

The collection of translation errors for the localities around the RCPs gives a picture of the parallelism of neighbouring sections of the trajectory in these local regions of the phase space. However, choosing a different set of RCPs results in a different set of translation errors. Thus, to minimize the stochastic error associated with choosing different sets of RCPs, the median translation error is taken for Q sets of RCPs and the mean of the Q medians is taken as the overall translation error of the trajectory [172]. Nonetheless, the values of M and Q are not found to have any significant effect.

Although the overall translation error of a trajectory indicates the degree of determinism of the dynamics of the system for the total time during which the system is observed, it would be more useful to know how the determinism of the system dynamics changes in time. This can be done using a running window on the time series and calculating the translation error as described above in the window. The N data points that make up the time series $s(n)$ are divided into a number of windows of equal size. The window size can either be specified by the number of data points N_w contained within it or by its temporal width $dt = N_w(1/f_s)$. The translation error and the midpoint of the window form coordinate pairs that are plotted to give the time evolution of the translation error. Applying a window overlap gives a smooth trace and is especially useful for short time series.

The above procedure is applied to a synthetic signal that is composed of deterministic and stochastic components. The deterministic component is the time series of x_1 from the Lorenz system. The stochastic component is a time series ξ of white Gaussian noise with a variance σ^2 that is a sinusoidal function of time. The time dependence of σ^2 is given in equation 2.13, where

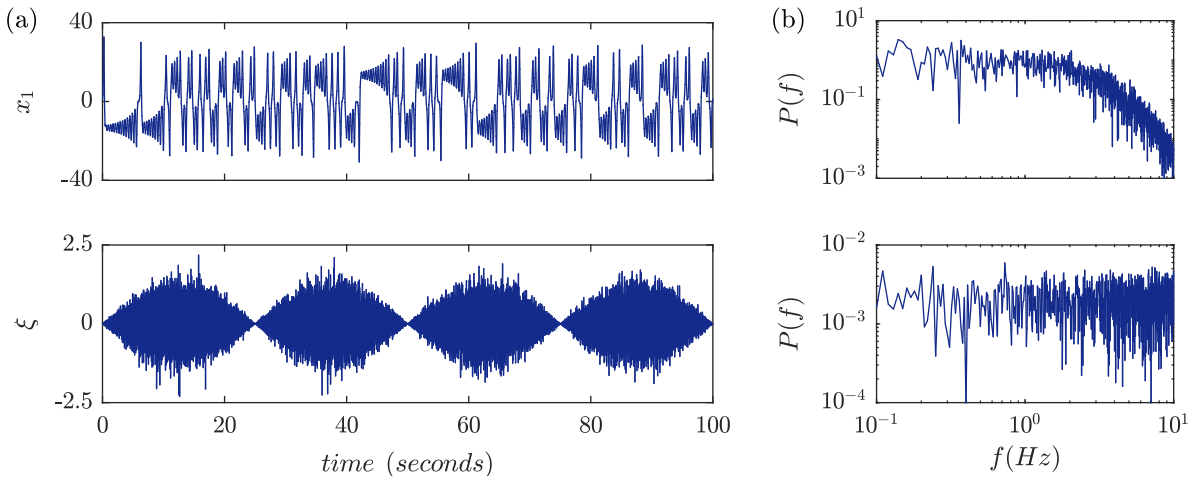

Figure 2.6: Synthetic signal: (a) time series, and (b) power spectra.

$|sin()|$ denotes the absolute value of the sinusoid, $A_\sigma = 0.5$ is the peak value, and $f_\sigma = 2$ is the frequency. The time series of x_1 and ξ are shown in figure 2.6a. The frequency spectra of the time series are shown in figure 2.6b. The x_1 and ξ time series are aperiodic and have broadband frequency spectra, which are characteristics of random time series. However, only the ξ time series is truly random. The x_1 time series is chaotic, that is the apparent randomness is due to the nonlinearity in the governing equations. In phase space the underlying dynamics of the x_1 time series are structured and that of the ξ time series are not. It should be noted that the regularity of the ξ time series in figure 2.6a is due to its periodic variance. The periodic variance is added intentionally to vary the determinism of the composite signal. Hence, when the variance is crossing zero the composite signal is mostly deterministic and when the variance goes round its peak value the composite signal is less deterministic. In phase space, the stochastic component disturbs the smooth trajectory of the deterministic component. These periodic disturbances in the phase space should be detected by plotting the translation error for the composite signal using a running window.

$$\sigma^2 = |A_\sigma sin(2\pi f_\sigma t)| \tag{2.13}$$

The composite signal $x_1 + \xi$ is shown in figure 2.7a. For the calculation of the translation error, a running window of width $dt = 1s$ is used with 90% overlap. The trajectory in phase space during the interval captured in the window is reconstructed using time-delay embedding with a time delay $T = 96$ and an embedding dimension $D = 3$. The trajectory for the interval $t = 47.5s - 52.5s$ is shown in figure 2.7b. During this interval, the variance of the stochastic component is close to zero as can be seen in the plot of ξ in figure 2.6a and hence, the determinism of the signal is at its maximum. Thus, the trajectory is smooth and neighbouring sections of the trajectory exhibit a good degree of parallelism. On the contrary, the trajectory for the interval $t = 60s - 65s$ that is also shown in figure 2.7b, when the determinism of the signal is at a minimum, is not smooth and neighbouring sections of the trajectory are not parallel. Hence,

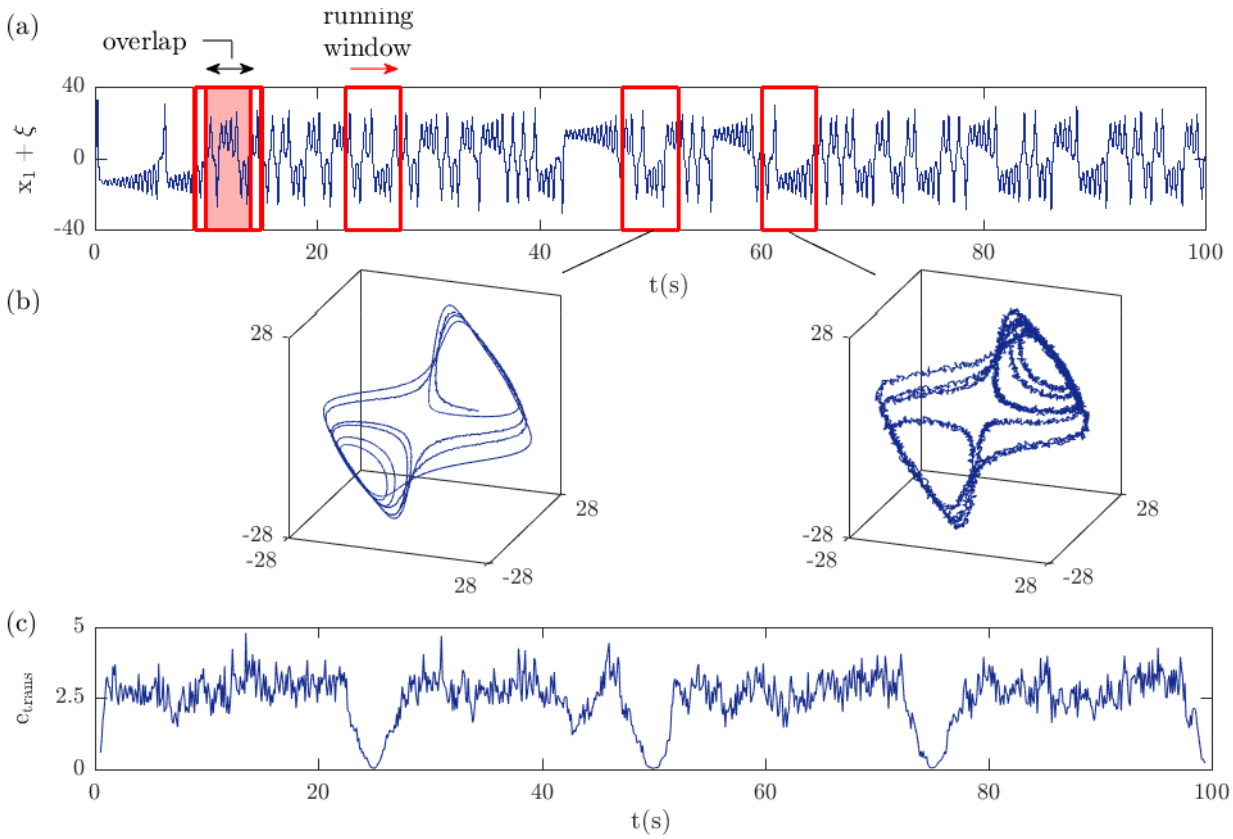

Figure 2.7: Synthetic data: (a) composite signal, (b) phase portraits reconstructed for two windows of the signal when the translation error is at its minimum and maximum, and (c) the translation error that is calculated using a running window of width $dt = 1s$ (not drawn to scale) with 90% overlap and parameters $T = 96$, $D = 3$, $Q = 3$, $M = 25$, $K = 3$.

the degree of parallelism of neighbouring sections of the trajectory is being influenced by the strength of the stochastic signal. As the translation error is a measure of the degree of parallelism of neighbouring sections of the trajectory its value should change in synchronisation with the variance of the stochastic component. The translation error is calculated using $Q = 3$ sets of $M = 25$ RCPs and $K = 3$ for every instance of the running window. The translation error is shown in figure 2.7c. It clearly changes periodically at the same frequency as does the variance of the stochastic signal shown in figure 2.6a. Therefore, it is clearly exemplified that the translation error can be used to detect changes in the dynamics of a system.

The MATLAB code written and used in the current work to calculate the translation error of a trajectory using a running window with overlap, given a time series, is included in appendix A.1.1.

2.1.4 Recurrence plots (RPs)

A fundamental characteristic of conservative systems is that the state of the system often recurs [142]. A state recurs when the trajectory of the system returns to a previously visited region

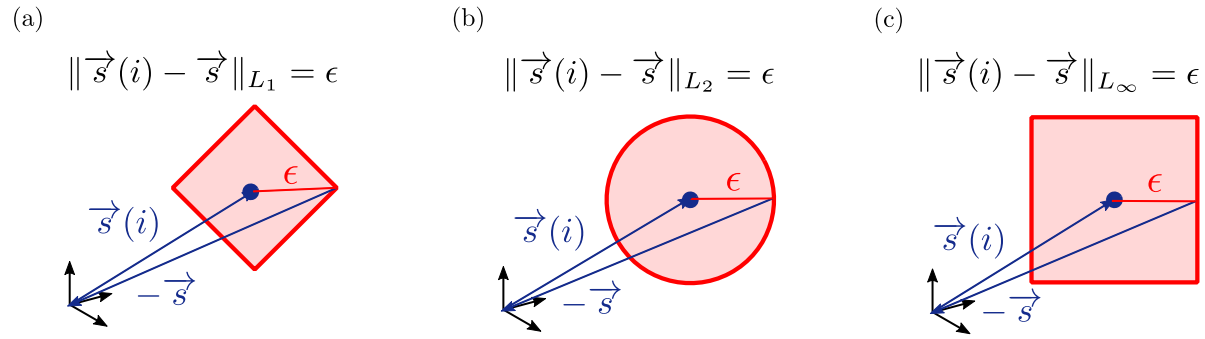

Figure 2.8: Neighbourhoods of the same search radius ε using various vector norms: (a) L_1-norm, (b) L_2-norm, and (c) L_∞-norm.

in phase space. Eckmann et al. [44] introduced the method of recurrence plots to visualize the recurrences of a system. The patterns that appear in the recurrence plots are related to specific dynamical behaviour. Hence, analysis of the patterns in these plots can provide qualitative information about the underlying dynamics of the system.

The state $\vec{s}(n)$ of a system does not always recur exactly, that is, the trajectory does not always return to the exact same point in phase space but comes very close to it. Hence, to identify recurrences it is necessary to search a neighbourhood of radius ε around each phase point $\vec{s}(i)$ on the trajectory. The distance between a phase point $\vec{s}(i)$ whose neighbourhood is being searched and another phase point $\vec{s}(j)$ is the magnitude of the vector $\vec{s}(i)$-$\vec{s}(j)$ that connects the two phase points. The magnitude of the vector can be determined using various norms and the norm that is used determines the shape of the neighbourhood. The vector norms that are commonly used are the L_1-norm, the Euclidean norm L_2, and the infinity norm L_∞. The shape of the neighbourhood of radius ε for these commonly used norms are shown in two-dimensional phase space in figure 2.8. The current work uses the L_2-norm and thus, the neighbourhood is circular in two-dimensional phase space and spherical in three-dimensional phase space. The radius ε of the neighbourhood is termed the recurrence threshold. The selection of an appropriate recurrence threshold can be made using several approaches [127]. In the current work the threshold is chosen between 3% and 5% of the maximum attractor diameter which is the maximum distance between any two phase points on the trajectory.

Recurrences are recorded in the recurrence matrix $R_{i,j}$ using equation 2.14, where Θ is the Heaviside step function, $\vec{s}(i)$ is the phase point whose neighbourhood is being searched, $\vec{s}(j)$ are the phase points on the trajectory whose distance from $\vec{s}(i)$ is to be measured, and $i, j = 1, \ldots, N$ with N being the total number of phase points on the trajectory. The distance $\|\vec{s}(i) - \vec{s}(j)\|$ between the phase point $\vec{s}(i)$ whose neighbourhood is being searched and another phase point on the trajectory $\vec{s}(j)$ is subtracted from the recurrence threshold ε and the sign of the difference determines whether the phase point $\vec{s}(j)$ is in the neighbourhood of $\vec{s}(i)$ or not. In figure 2.9a, the difference $\varepsilon - \|\vec{s}(i) - \vec{s}(j)\| \geq 0$ is equivalent to $\varepsilon \geq \|\vec{s}(i) - \vec{s}(j)\|$. Hence, the phase point $\vec{s}(j)$ is in the neighbourhood of the phase point $\vec{s}(i)$ and the Heaviside step

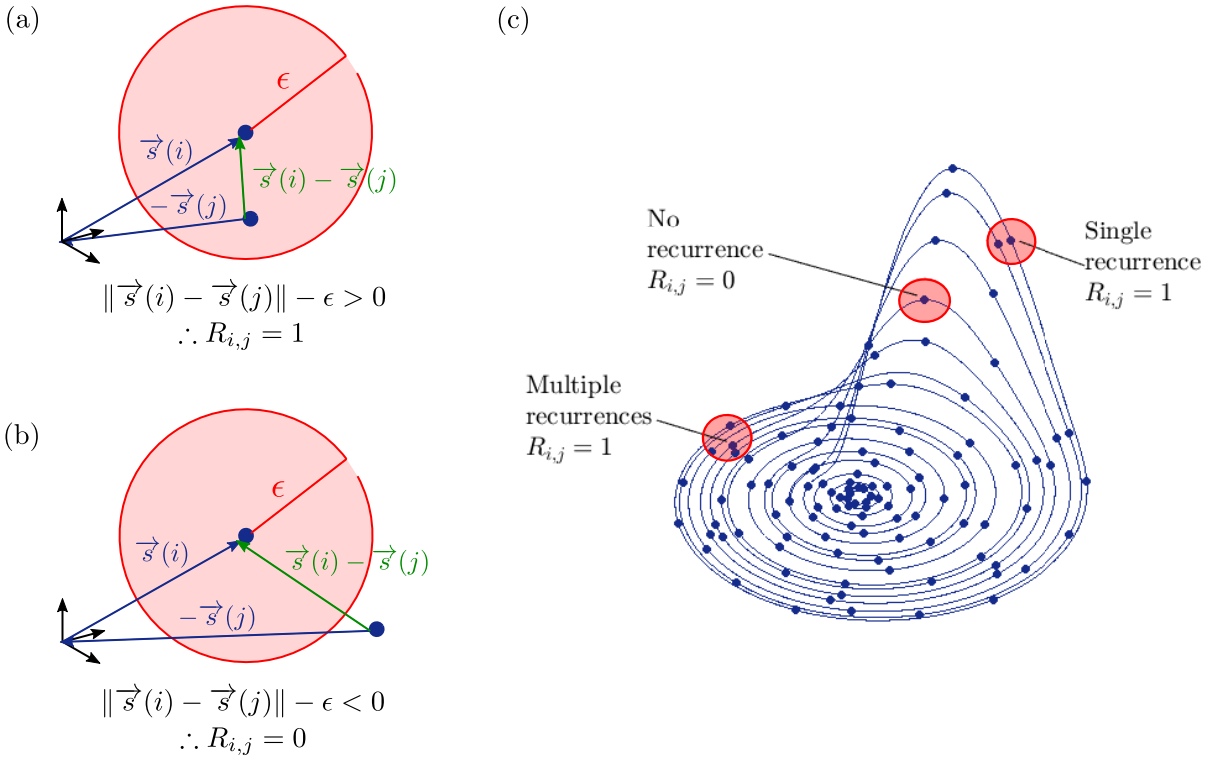

Figure 2.9: Identifying recurrences in phase space: (a) state $\vec{s}(j)$ is a recurrence of $\vec{s}(i)$, (b) state $\vec{s}(j)$ is not a recurrence of $\vec{s}(i)$, and (c) recurrences in the phase portrait of the Rossler system with parameters $a = 0.15, b = 0.2$, and $c = 10$ constructed using a time step $dt = 0.01$ and $10,000$ points (for clarity only a fraction of the total points are shown).

function of the positive argument gives $R_{i,j} = 1$. In figure 2.9b, the difference $\varepsilon - \|\vec{s}(i) - \vec{s}(j)\| < 0$ is equivalent to $\varepsilon < \|\vec{s}(i) - \vec{s}(j)\|$. Hence, the phase point $\vec{s}(j)$ is not in the neighbourhood of the phase point $\vec{s}(i)$ and the Heaviside step function of the negative argument gives $R_{i,j} = 0$. In figure 2.9c, example recurrences are shown for a trajectory of the chaotic Rossler system [127].

$$R_{i,j}(\varepsilon) = \Theta\left(\varepsilon - \|\vec{s}(i) - \vec{s}(j)\|\right) \tag{2.14}$$

The recurrences of the trajectory can be visualised in a plane with coordinates i, j by plotting a dot at every i, j at which $R_{i,j} = 1$. The resulting plot is the recurrence plot (RP) of the trajectory. The RP of the deterministic simple harmonic oscillator, the chaotic Rossler system, and stochastic white Gaussian noise are shown in figure 2.10. The recurrences of these dynamically very different systems form different patterns in the RPs. However, certain features of RPs are universal and independent of the system dynamics. Firstly, every phase point is a recurrence of itself. That is, $R_{i,j} = 1$ for every $i = j$ and hence, an RP always contains a main diagonal line that is termed the line of identity (LOI). The LOI is found in all the RPs shown in figure 2.10. Secondly, when the same ε is used for every phase point $\vec{s}(i)$ the resulting RP is symmetric about the LOI because $R_{i,j} = R_{j,i}$. In other words, if $\vec{s}(j)$ falls in the neighbourhood of $\vec{s}(i)$ then $\vec{s}(i)$ will fall in the neighbourhood of $\vec{s}(j)$ as long as the neighbourhoods are the same

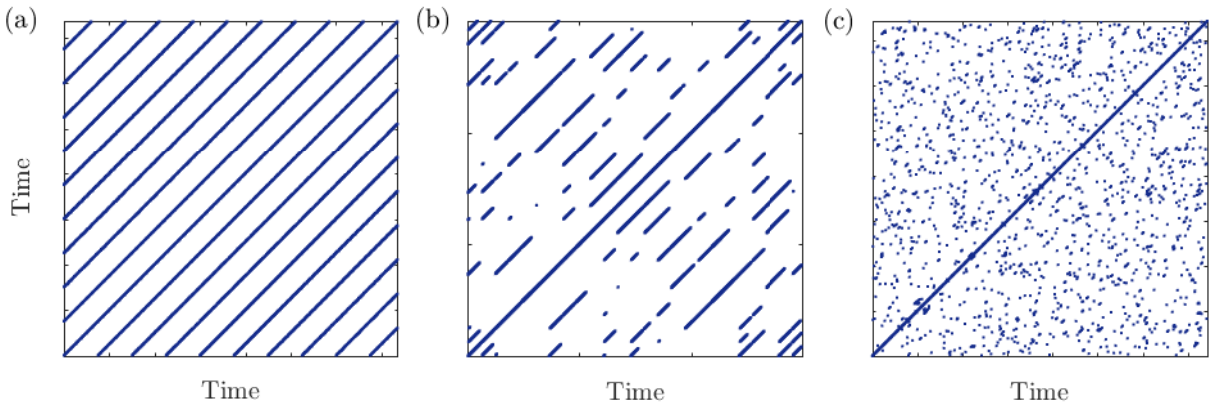

Figure 2.10: Signature structures in recurrence plots: (a) the simple harmonic oscillator, (b) the chaotic Rossler system, and (c) white Gaussian noise.

size. This universal feature is also found in all the RPs shown in figure 2.10.

Small scale and large scale structures in RPs are related to specific dynamic behaviours [127]. Diagonal lines are of particular interest because they are a sign of deterministic dynamics and more specifically of periodic behaviour. This is exemplified by the RP of the simple harmonic oscillator shown in figure 2.10a. The initial state of the simple harmonic oscillator corresponds to the dot at the origin of the axis in figure 2.10a, which simply indicates the trivial recurrence of the initial state with itself. After the simple harmonic oscillator has been set in motion it will return to its initial state after time equal to one period of the motion. Hence, there are equidistant dots along the vertical axis of the RP and since the RP is symmetric about the LOI, similarly there are equidistant dots along the horizontal axis. The distance between the dots is the period of the periodic motion. Therefore, the dots on the axis of the RP in figure 2.10a are the recurrences of the initial state of the simple harmonic oscillator. The state of the system immediately after the initial state also recurs with itself and is the next dot along the LOI. This state also recurs periodically and the recurrences of this state are equidistant dots along the vertical and horizontal that are shifted in the direction in which time flows. The recurrences of later states appear in the RP in the same way and the dots corresponding to their recurrences shift to the right and the diagonal lines form in the RP and progressively increase in length as long as the states keep recurring. In the event that the states stop recurring in time and hence, the periodicity of the system is disrupted, the elongation of the diagonal lines in the RP is also disrupted. The RP of white Gaussian noise that is shown in figure 2.10c has very short if any diagonal lines, thus confirming that there is no periodicity and the states of the system in this case are completely uncorrelated. The RP of the chaotic Rossler system shown in figure 2.10b contains diagonal lines of various lengths and single isolated points.

Other structures such as horizontal and vertical lines that appear in RPs can be associated with specific dynamical behaviour of the system. Typical patterns and their meanings are summarised in table 2.1.

Table 2.1: Typical structures found in recurrence plots and the associated dynamical behaviour that they represent [127].

Pattern	Interpretation
Homogeneity	The process is stationary
Fading to the upper left and lower right corners	Non-stationary data; the process contains a trend or a drift
Disruptions (white bands)	Non-stationary data; some states are rare or far from normal; transitions may have occured
Periodic/quasi-periodic patterns	Cyclicities in the process; the time distance between periodic patterns (e.g. lines) corresponds to the period; different distances between long diagonal lines reveal quasi-periodic processes
Single isolated points	Strong fluctuation in the process; if only single isolated points occur, the process may be uncorrelated random or even anti-correlated process
Diagonal lines (parallel to the LOI)	The evolution of states is similar at different epochs; the process could be deterministic; if these diagonal lines occur beside single isolated points, the process could be chaotic (if these diagonal lines are periodic, unstable periodic orbits can be observed)
Diagonal lines (orthogonal to the LOI)	The evolution of states is similar at different times but with reverse time; sometimes this is an indication of an insufficient embedding
Vertical and horizontal lines/-clusters	Some states do not change or change slowly for some time; indication for laminar states
Long bowed line structures	The evolution of states is similar at different epochs but with different velocity; the dynamics pf the system could be changing

The MATLAB code written and used in the current work to generate the recurrence matrix, given a time series, is included in appendix A.1.1.

2.1.5 Recurrence quantification analysis (RQA)

Recurrence quantification analysis (RQA) introduces statistical measures that exploit some of the observations summarized in table 2.1 to quantify underlying dynamics. The line of identity (LOI) is not included in the calculation of the RQA measures.

The determinism DET [2] is defined in equation 2.15, where l is the length or number of dots that form the diagonal lines, l_{min} is a threshold diagonal line length, and $P(l)$ is the histogram of diagonal lines of length l. The DET measure is simply the ratio of the number of dots in the RP that form diagonal lines to the total number of dots. The threshold diagonal line length l_{min} is defined to avoid counting false recurrences, such as those associated with diagonal lines that are a consequence of tangential motion. Tangential motion refers to the situation where a recurrence is recorded falsely as a consequence of the neighbourhood ε encapsulating one or more phase points that are before or after the current phase point whose neighbourhood is being searched and are on the same trajectory segment. Tangential motion can be the result of using a large ε i.e. searching a large area and/or using a time series that has been sampled at high frequency in which case the phase points in the reconstructed phase space are densely packed along the trajectory. The present work uses a minimal length for diagonal lines that is $l_{min} = 2$, which is that recommended in the study of Babaei et al. [6]. In addition to DET [2] the RQA measures RR [2] and DIV [2] are also calculated in the present work and their definitions are given below.

$$DET = \frac{\sum_{l=l_{min}}^{N_{ps}} lP(l)}{\sum_{l=1}^{N_{ps}} lP(l)} \tag{2.15}$$

The recurrence rate RR [2] is defined in equation 2.16, where $R_{i,j}$ is the recurrence matrix. The RR is the density of recurrence points in the RP. A more physical interpretation is that it is the probability of the dynamical system returning to a previous state.

$$RR = \frac{1}{N_{ps}^2} \sum_{i,j=1}^{N_{ps}} R_{i,j} \tag{2.16}$$

The divergence DIV [2] is defined in equation 2.17, where L_{max} is the length of the longest diagonal line in the RP excluding diagonal lines of length $l < l_{min}$. Thus, DIV is the reciprocal of the length of the longest diagonal line in the RP. It indicates how fast the trajectories diverge in phase space. A small L_{max} is an indication that the trajectories in phase space diverge very fast and this results in DIV being large. On the other hand if L_{max} is large then the trajectories

diverge slowly from each other and DIV is small.

$$DIV = \frac{1}{L_{max}} \tag{2.17}$$

The MATLAB code written and used in the current work to calculate the above RQA measures using a running window with overlap, given a time series, is included in appendix A.1.1.

2.1.6 Dynamical properties of actual and reconstructed trajectories

The translation error and recurrence quantification analysis measures RR, DET, and DIV are calculated for the actual and the reconstructed trajectories of the Lorenz system in equation 2.8 to demonstrate that these have the same properties. For the purpose of using these metrics to identify changes in the dynamical behaviour of a system, it is important that their evolution for the reconstructed system is the same with that of the actual system. The actual values of the metrics are not important in this respect. Nonetheless, it is shown that the values of the metrics for the reconstructed system can be in good agreement with those of the actual system.

The actual trajectory of the Lorenz system is shown in figure 2.11a and consists of $100,000$ phase points. The coordinates of the points are the states of the Lorenz system that are obtained by solving the Lorenz equations ($\sigma = 16$, $b = 4$, and $r = 45.92$) using the Runge-Kutta method and a time step $dt = 0.001$. The reconstructed trajectory is shown in figure 2.11b and consists of $99,424$ phase points. In this case, the coordinates of the phase points are time-delayed (with $T = 96$ and $D = 3$) values of the x_1 state of the Lorenz system. The translation error and the RQA measures RR, DET, and DIV of the actual and reconstructed trajectories are shown in figure 2.11c-f. The translation error of the trajectories is calculated according to the methodology in section 2.1.3 using a running window that is one second wide with no overlap and the parameters $M = 250, Q = 5, k = 3$. The RQA measures are calculated according to the methodology in section 2.1.5 using a running window with the same width and no overlap and the parameter $\varepsilon = 0.03$.The values of the parameters is not of great importance since the same values are used to calculate the metrics for both the actual and reconstructed trajectories.

In figure 2.11c, there is a large offset between the translation error of the reconstructed trajectory and that of the actual trajectory, the former being $\approx 200\%$ larger. However, both values are extremely close to zero and thus, the translation error of the reconstructed trajectory indicates, just as the translation error of the actual trajectory does, that the dynamics of the system are completely deterministic. It is not surprising that the values are not exactly the same since the reconstructed trajectory is an approximation of the actual trajectory. Nonetheless, both values indicate strongly deterministic dynamics. More importantly for the needs of the current work, there is no disagreement regarding the evolution of the translation error.

The values and evolution of the recurrence rate (RR) in figure 2.11d and the determinism (DET) in figure 2.11e of the reconstructed and actual systems are in perfect agreement. The

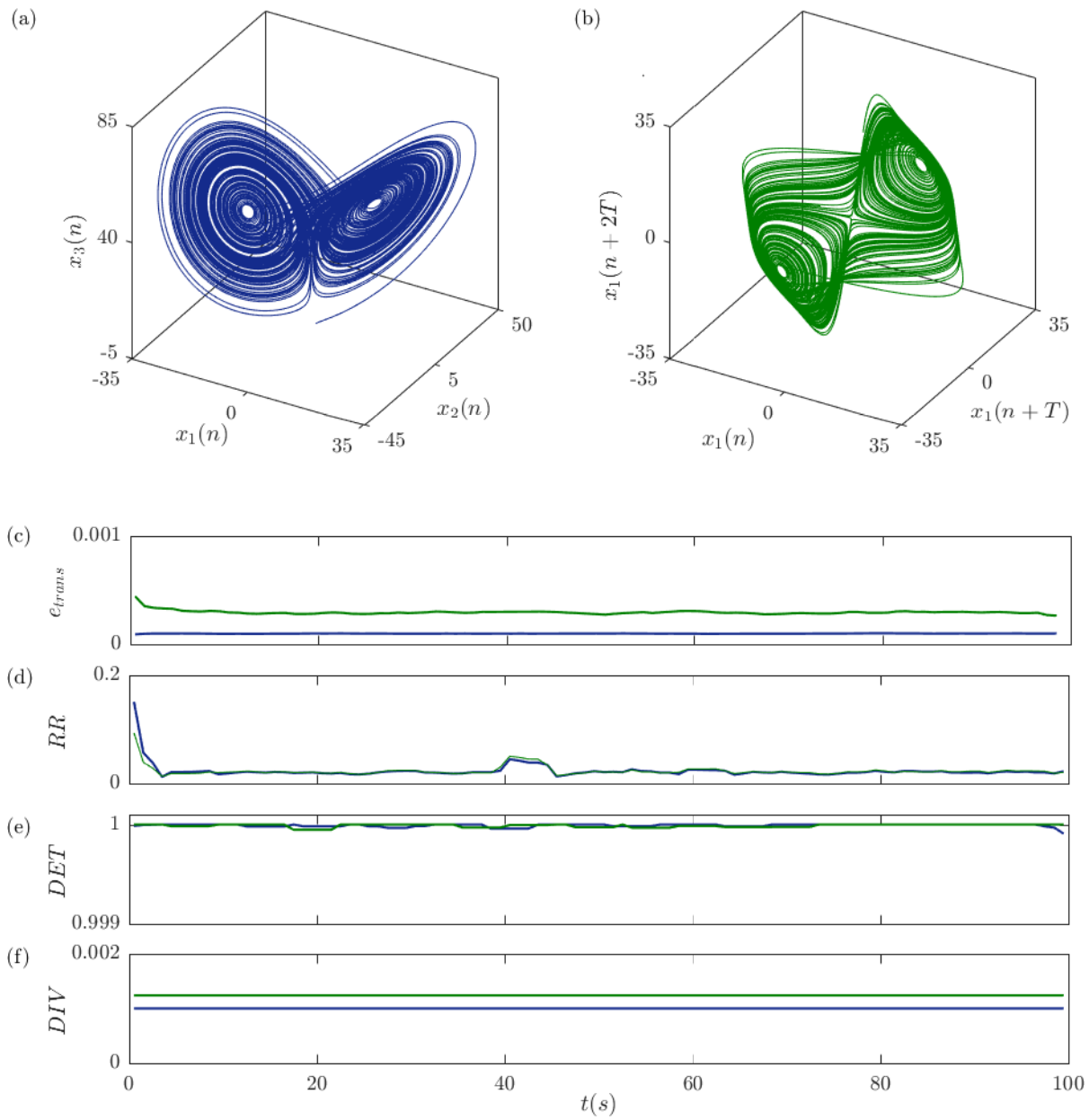

Figure 2.11: Comparison of the translation error and the RQA measures of the actual and reconstructed Lorenz system with $\sigma = 16$, $b = 4$, and $r = 45.92$: (a) the actual trajectory with 100,000 points, (b) the reconstructed trajectory by time-delay embedding the time series of $x_1(t)$ using $T = 96$ and $D = 3$, (c) the evolution of the translation error, (d) the evolution of RR, (e) the evolution of DET, and (f) the evolution of DIV.

DET is unity, which is the maximum value, indicating that the systems are completely deterministic. This is in agreement with the indication of the value of the translation error in figure 2.11c, which also indicates that the dynamics are completely deterministic.

In figure 2.11f, the divergence (DIV) of the reconstructed system is $\approx 25\%$ larger than that of the actual system. That is, the reconstructed trajectory diverges from neighbouring sections of the trajectory faster than the actual trajectory. However, more importantly for the needs of the current work, the evolution of DIV is the same.

2.2 Direct numerical simulation (DNS) method

The simulation is performed using a variant of the Imperial College in-house flow solver with the acronym BOFFIN (BOundary Fitted Flow INtegrator) [79, 124, 140, 4]. The code is written in Fortran 77/95 with MPI subroutines for multiprocessor architectures. The governing equations and numerical procedures of the flow solver BOFFIN are discussed briefly below.

2.2.1 Governing equations

Fluid flow is governed by equations 2.18*a-c*, where the dependent variables ρ, u_i, p, and T are density, velocity, pressure, and temperature respectively. Derivatives of the dependent variables are with respect to time t and the spatial coordinates x_i. The constants c_p, k, and g_i are the specific heat at constant pressure, thermal conductivity, and the gravitational acceleration vector respectively. A repeated subscript implies summation over the index range $i = 1, 2, 3$.

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_j}{\partial x_j} = 0, \tag{2.18a}$$

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} + \rho g_i, \tag{2.18b}$$

$$\rho c_p \left(\frac{\partial T}{\partial t} + u_i \frac{\partial T}{\partial x_i} \right) = \frac{\partial p}{\partial t} + u_i \frac{\partial p}{\partial x_i} + k \frac{\partial^2 T}{\partial x_i \partial x_i} + \tau_{ij} \frac{\partial u_i}{\partial x_j}, \tag{2.18c}$$

The shear stress tensor τ_{ij} for a Newtonian fluid is related to the strain rate tensor S_{ij} through equation 2.19a, where μ is the dynamic viscosity of the fluid. The strain rate tensor S_{ij} is related to the velocity gradients through equation 2.19b. The thermodynamic properties of an ideal gas are related through equation 2.19c, where R is the specific gas constant.

$$\tau_{ij} - \frac{1}{3}\tau_{kk} = 2\mu \left(S_{ij} - \frac{1}{3} S_{kk} \right), \tag{2.19a}$$

$$S_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right), \tag{2.19b}$$

$$p = \rho R T. \tag{2.19c}$$

The flow solver BOFFIN is a low Mach number formulation and therefore, neglects the terms $\partial p/\partial t + u_i(\partial p/\partial x_i)$ and $\tau_{ij}(\partial u_i/\partial x_j)$ in equation 2.18*c*. The neglected terms are namely the total derivative of pressure and frictional heating respectively and are negligible at low Mach number [123]. Nonetheless, the flow solver is appropriate for the present study because flows through gas turbine combustors are at low Mach number (the bulk flow Mach number is in the range 0.1-0.2 [132]).

2.2.2 Discretization schemes

Space derivatives are discretised by a standard second order central difference scheme with a constant spatial step Δx according to equation 2.20 except for the convective term $\partial \rho u_i u_j / \partial x_j$ in the momentum equation 2.18b for which the energy conserving discretization scheme of Morinishi [133] is used.

$$\frac{\partial \mathbf{u}}{\partial x} = \frac{1}{2}\left(\frac{\mathbf{u}^{n}_{r+1} - \mathbf{u}^{n}_{r-1}}{\Delta x}\right) + O((\Delta x)^2) \tag{2.20}$$

Time derivatives are discretised by a second order accurate three point backward difference scheme with a constant time step Δt according to equation 2.21.

$$\frac{\partial \mathbf{u}}{\partial t} = \frac{3}{2}\left(\frac{\mathbf{u}^{n+1}_{r} - \mathbf{u}^{n}_{r}}{\Delta t}\right) - \frac{1}{2}\left(\frac{\mathbf{u}^{n}_{r} - \mathbf{u}^{n-1}_{r}}{\Delta t}\right) + O((\Delta t)^2) \tag{2.21}$$

2.2.3 Pressure projection method

The pressure and velocity fields are obtained by employing a SIMPLE-type [177] pressure correction method. This method is briefly explained below.

The finite difference equation of the u-momentum equation to within the second order accuracy can be written as in equation 2.22, where $\mathbf{u}$ is the vector of the unknown u nodal values, $\mathbf{T}^{n+1}$ is the finite difference coefficient of the convection and diffusion terms at $n+1$, $\mathbf{D}$ corresponds to the discretization of the pressure term, and the source term $\mathbf{S}$ contains all the remaining terms including the cross-stresses not contained within $\mathbf{T}^{n+1}$ which have been treated explicitly to reduce the computational cost.

$$\mathbf{u}^{n+1} - \mathbf{u}^{n} + \frac{2}{3}\Delta t \mathbf{T}^{n+1}\mathbf{u}^{n+1} = -\frac{2}{3}\Delta t \mathbf{D}\mathbf{p}^{n+1} + \mathbf{S} \tag{2.22}$$

The equation 2.22 is solved in a two stage process to maintain its second order accuracy. Firstly, the solution is sought at an intermediate time level m that is between the current time level n and the next time level $n+1$. Secondly, the solution at the intermediate time level is used to obtain the solution at the time level $n+1$.

The intermediate time level is introduced through a pressure increment $\Delta \mathbf{p}^{m} = \mathbf{p}^{m} - \mathbf{p}^{n}$ so that equation 2.22 becomes equation 2.23, where m represents an intermediate time level between n and $n+1$.

$$\left(\mathbf{I} + \frac{2}{3}\Delta t \mathbf{T}^{n}\right) \underbrace{\left(\mathbf{u}^{m} + \frac{2}{3}\Delta t \mathbf{D}\Delta \mathbf{p}^{m}\right)}_{\mathbf{u}^{*}} = -\frac{2}{3}\Delta t \mathbf{D}\mathbf{p}^{n} + \mathbf{S} \tag{2.23}$$

The solution of equation 2.23 is obtained in the two steps given in equations 2.24 and 2.25.

$$\mathbf{u}^{*} = \left(\mathbf{I} + \frac{2}{3}\Delta t \mathbf{T}^{n}\right)^{-1} \left(-\frac{2}{3}\Delta t \mathbf{D}\mathbf{p}^{n} + \mathbf{S}\right) \tag{2.24}$$

$$\mathbf{u}^{m} = \mathbf{u}^{*} - \frac{2}{3}\Delta t \mathbf{D}\Delta\mathbf{p}^{m} \tag{2.25}$$

However, none of these velocity fields at time level m can be obtained since $\Delta\mathbf{p}^{m}$ is not known. In order to obtain the pressure increment $\Delta\mathbf{p}^{m}$, the velocity fields $\mathbf{u}^{m}$, $\mathbf{v}^{m}$ and $\mathbf{w}^{m}$ are substituted into the continuity equation, giving a Poisson like equation for the pressure increment. Since $\mathbf{u}^{m}$ is a second order accurate approximation to $\mathbf{u}^{n+1}$, in the second stage, a second order accurate solution at the time level $n+1$ is obtained by rewriting equation 2.23 for the time level $n+1$, with the coefficient matrix is evaluated using the most recent update for the flow field obtained at time level m.

$$\left(\mathbf{I} + \frac{2}{3}\Delta t \mathbf{T}^{m}\right) \underbrace{\left(\mathbf{u}^{n+1} + \frac{2}{3}\Delta t \mathbf{D}\Delta\mathbf{p}^{n+1}\right)}_{\mathbf{u}^{**}} = -\frac{2}{3}\Delta t \mathbf{D}\mathbf{p}^{m} + \mathbf{S} \tag{2.26}$$

Again, equation 2.26 is solved in the two steps given in equations 2.27 and 2.28.

$$\mathbf{u}^{**} = \left(\mathbf{I} + \frac{2}{3}\Delta t \mathbf{T}^{m}\right)^{-1} \left(-\frac{2}{3}\Delta t \mathbf{D}\mathbf{p}^{m} + \mathbf{S}\right) \tag{2.27}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^{**} - \frac{2}{3}\Delta t \mathbf{D}\Delta\mathbf{p}^{n+1} \tag{2.28}$$

Using the above mentioned pressure correction algorithm, the computed pressure and the velocity components are stored at the centre of a control volume according to the collocated grid arrangement. The Poisson like pressure correction equation is discretised by using the Rhie and Chow [148] pressure smoothing approach, which prevents the even–odd nodes uncoupling in the pressure and velocity fields. The BI-CGSTAB [169] solver is used for solving the matrix of velocity vectors, while for the Poisson like pressure correction equation an ICCG [93] solver is applied due to its symmetric and positive definite nature.

2.2.4 Convective thermal boundary condition

The original code is limited to a constant temperature thermal boundary condition which is not representative of the thermal condition at the walls of a gas turbine combustor. In real combustors cooling air flows externally around the walls. Hence, the current work adds a convective thermal boundary condition to the DNS code BOFFIN according to which the temperature at the walls satisfies the heat flux equality in equation 2.29, where k_{fluid} is the thermal conductivity of the fluid which is assumed to be a constant, $(\partial T/\partial y)|_{wall}$ is the temperature gradient at the wall,

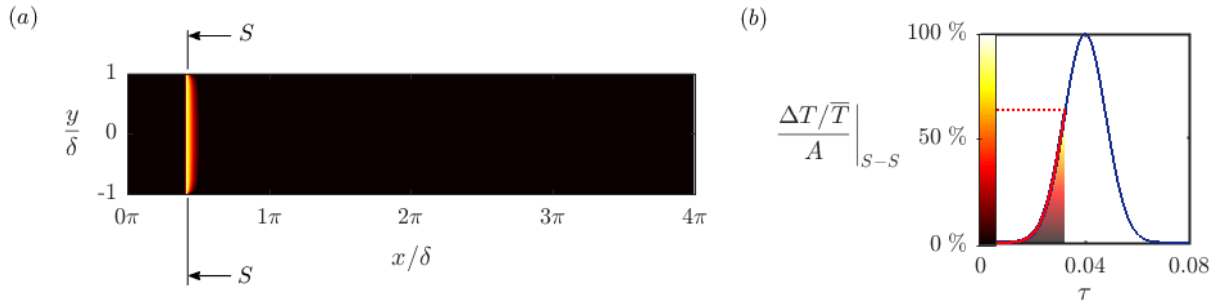

Figure 2.12: Entropy wave generation: (a) the temperature is perturbed in a cross-section immediately downstream of the channel inlet, and (b) the temporal profile of the perturbation is Gaussian as per equation 2.31 with $A = 0.1$, $\sigma = 0.1$, and $\mu = 0.5$.

T_{wall} is the wall temperature, and h is the convective heat transfer coefficient between the wall and a imaginary external flow (not simulated) at temperature T_∞. The l.h.s of equation 2.29 is the heat flux from the stationary (no-slip) fluid layer to the wall in accordance to Fick's law and the r.h.s is the heat flux from the wall to the hypothetical external flow according to Newton's law. The wall is assumed to be very thin and thus, conduction of heat through the wall itself is not considered. This thermal boundary condition enables the specification of adiabatic walls through setting $h = 0 W m^{-2} K^{-1}$ and streamwise varying heat transfer through setting a finite h.

$$k_{fluid} \left.\frac{\partial T}{\partial y}\right|_{wall} = h\left[T_{wall}(x) - T_\infty\right] \tag{2.29}$$

2.2.5 Temperature forcing

The entropy of an ideal gas is related to temperature and pressure through the thermodynamic equation 2.30, where s is entropy. The pressure term dp/p in equation 2.30 is usually small and entropy fluctuations therefore scale with temperature fluctuations [52, 131, 132]. Hence, an entropy wave is added to the flow by perturbing the temperature of the base flow in a cross-section immediately downstream of the channel inlet. The method resembles that of the Entropy Wave Generator (EWG) used in the experimental studies [8, 9] of entropy noise, which, adds entropy waves to an accelerating tube flow by means of a heating module.

$$\frac{ds}{c_p} = \frac{dT}{T} - \frac{R}{c_p}\frac{dp}{p} \tag{2.30}$$

The temporal profile chosen for the temperature perturbation is Gaussian as per equation 2.31, where T is instantaneous temperature and therefore $\Delta T/\overline{T} = (T - \overline{T})/\overline{T}$ is the instantaneous amplitude of the perturbation relative to the mean flow temperature $\overline{T}$, A is the peak amplitude of the perturbation, τ is non-dimensional time, μ is the non-dimensional time at which the perturbation reaches its peak amplitude, and σ is the standard deviation of the perturbation that controls the duration of the temperature forcing. The non-dimensional time is defined as $\tau = (\overline{U}_{bulk}/L)t$, where $\overline{U}_{bulk}$ is the bulk flow velocity, L is the channel length and therefore

non-dimensional time is in multiples of the mean residence time. At every time step the instantaneous amplitude of the perturbation is uniform over the plane cross-section at which it is added. The present study uses $A = 0.1$, $\sigma = 0.1$, and $\mu = 0.5$. The process of generating the entropy wave is illustrated in figure 2.12. The graph in the left column of figure 2.12 shows the Gaussian perturbation used in the present study, where the instantaneous amplitude of the perturbation is plotted with respect to non-dimensional time τ. In the graph it can be seen that the time taken to add the entropy wave to the flow is approximately one half of the mean residence time.

$$\frac{\Delta T}{\overline{T}} = A\ exp\left[-\frac{(\tau-\mu)^2}{2\sigma^2}\right] \tag{2.31}$$

Chapter 3

Precursors of flame flashback in unstable and stable combustion

3.1 Introduction

Flame flashback is a concerning issue for the emerging lean premixed combustion technology that promises enhanced system efficiency and significant reduction in pollutant emission. Not only has the phenomenon been impossible to detect in advance, the complexity and the highly transient nature of the phenomenon make investigations intractable. Thus, understanding and suppression of this intricate issue has been a long standing issue and continues to challenge the research community as well as the industry.

Flame flashback is a complex phenomenon resulting from interactions involving multiple physical processes, each governed by its own inherent temporal and spatial scale. In addition, flashback is also a transient event, with the actual event lasting only fractions of a second. Investigation of flashback requires multiple, simultaneous diagnostics without prior knowledge of the relevant time and length scales of the physical processes involved. Detection, accordingly, deals with post-event characterization. In this chapter, an attempt to detect subtle dynamics prior to flame flashback—which linear, time and frequency-domain methods cannot reveal—is made through recurrence analysis of the multidimensional phase space reconstruction from pressure time traces. Recurrence analysis is known to be superior to other techniques for short, noisy datasets. Following conventional recurrence analysis methods, different sections of pressure time traces from reacting flow conditions are analysed. Recurrence plots and recurrence quantification measures immediately before flashback are compared to those long before flashback during the normal operation of the combustor. A comparison between recurrence characteristics at isothermal flow conditions and at reacting flow conditions is included to a) highlight the dynamics of the thermoacoustically unstable condition at which flashback occurs and b) to establish the limits of fluctuations in recurrence quantification analysis measures.

Flashback refers to the unforeseen and instantaneous propagation of the reactive front to the

upstream stages of the combustor [108, 12, 114]. Because upstream stages are not designed for the extreme conditions created by the flame, a flashback event can cause critical damages that must be avoided by system shut-down. Thus, the phenomenon is clearly undesirable and detecting the onset of flame flashback under various operating conditions, is an essential step in the design of premixed combustion systems.

Flashback can happen in the boundary layer, in the core of a swirling flow, due to combustion induced vortex breakdown and due to combustion instabilities [155, 56]. In swirling flows the flame can propagate upstream on the vortex axis [76]. A popular theory that explains the rapid upstream flame propagation along the vortex axis is the theory of vortex bursting [24]. According to the vortex bursting theory the pressure downstream of the flame increases and the resultant force from the pressure imbalance across the flame pushes the flame upstream. Regardless of the responsible mechanism, flame flashback is always a highly transient phenomenon, which takes a small fraction of a second for completion. This transient nature of flashback makes it exceedingly difficult to control and suppress. As a result, an early detection of the possibility of flashback is of high significance. In a recent study, Gotoda et al. [63] proposed the use of the translation error as an online method for detecting a precursor of lean blowout.

The primary cause of flashback is the imbalance between the local flame and flow velocities. In a practical setting that involves highly turbulent flows, flame interaction with hydrodynamics (boundary layer [13, 47, 49] and coherent structures [155, 88, 98, 99]), acoustics (thermoacoustic coupling) [168, 91, 165], chemical kinetics [125, 12] and unsteady heat transfer among other physical processes, each with an inherent time and length scale, the said imbalance—hence, sudden flashback—can be caused due to several factors. The referred works deal with the study of how the respective factors affect the flashback propensity of the system and/or the evolution of the individual processes as flashback takes place. It has been made clear that the physical mechanisms responsible for flame flashback are quite complex and are not yet sufficiently understood so as to efficiently eliminate the possibility of flashback. The transition from one dynamical state, prior to flashback, to a totally different one corresponding to the state in which the flame is inside the premixing section can happen over an extremely short time (often a fraction of a second). Understanding the dynamics of such a transitional period is essential to the implementation of suppression mechanisms.

Due to the short temporal span of the transitional period, conventional time and frequency domain methods such as spectral analysis, become inaccurate in analysing the system. Such techniques are inherently incapable of detecting the subtle complex features of the dynamics of flashback. Another problem associated with the highly complex and transient process is that without the prior knowledge of the temporal and spatial scales involved in flashback, in a given configuration, experimental diagnostics of the event [69, 100] are difficult to perform. Changes prior to flashback are not apparent on direct visual or quantitative examination of time series and its spectral decomposition. Identification of the presence of system dynamics—and changes

therein—that could be considered precursors to the event of flashback are, therefore, of immense practical and academic interest.

The work that follows identifies precursors of flashback in stable and unstable combustion in laboratory burners by analysing the trajectory of the systems in phase space, which is reconstructed from pressure time series measurements using time delay embedding [162].

3.2 Pressure time series from flashback experiments

The pressure time series that are used in the current work are from flashback experiments carried out at the Technical University of Darmstadt [88, 89] and at the University of Cambridge [10, 81, 82]. It should be made clear that the experiments are not part of the current work and will only be described briefly to put the current work into context.

The experiments were conducted in laboratory swirl burners with central bluff-body. Schematics of the burners are shown in figure 3.1. The swirl burner in figure 3.1a has a nozzle open to the atmosphere and a bluff-body with a cylindrical end that protrudes slightly from the nozzle exit plane. The experiment in this burner observed flame flashback in stable combustion. Flame flashback in this case was triggered by increasing the swirl number above a critical value. The swirl burner in figure 3.1b has a nozzle connected to a flame enclosure and a bluff-body with a conical end that is flush with the nozzle exit plane. The experiment in this burner observed flame flashback in thermoacoustically unstable combustion. Flame flashback in this case occurred while the mean flow properties were constant. Although in this burner flame flashback occurred in unstable combustion without any action from the experimentalist, it could not be confirmed that thermoacoustic instability is what caused it. Nonetheless, flame flashback was observed in two extremely different dynamical systems.

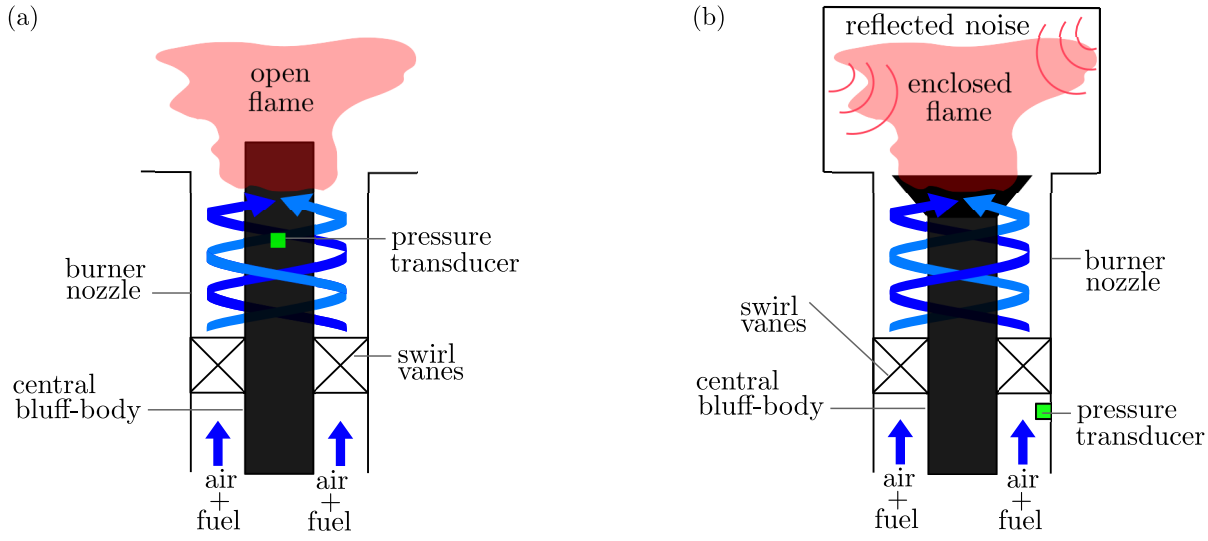

Figure 3.1: Schematics of the laboratory swirl burners: (a) the swirl burner at the Technische Universität Darmstadt [88, 89] and (b) the swirl burner at the University of Cambridge [81, 82].

Table 3.1: Pressure time series from the experiment of flashback in unstable combustion.

Pressure time series		U	ϕ^*
No flame	With flame*	(ms^{-1})	$(-)$
p_{10}	P_{10}	10	0.79
p_{12}	P_{12}	12	0.73
p_{14}	P_{14}	14	0.74

3.2.1 Thermoacoustically stable swirl burner experiment

At the start of the experiment in stable combustion, the flame is on the edge of the bluff-body. Recirculation zones in the wake of the bluff-body anchor the flame at this position. The swirl number is increased until it is above a critical value and the flame flashback occurs. Once the flame is visually confirmed to have moved upstream, the fuel supply is cut-off. The experiment is repeated four times with the same flow velocity and swirl number to produce four sets of pressure time series.

The pressure fluctuations were sampled at a sampling frequency of $f_s = 10kHz$ from three pressure transducers on the surface of the bluff-body (only one pressure transducer is shown in figure 3.1a). Since the pressure time series from the three locations are similar, as would be expected from locations that are close to each other, the analysis of the current work is only applied to the pressure time series from one of the transducers.

3.2.2 Thermoacoustically unstable swirl burner experiment

In the flashback experiments in unstable combustion, the average flow properties are kept constant during the experiment. The thermoacoustically unstable combustion is followed by flame flashback and subsequent stabilization of the flame in the swirler. As in the case of the experiment of flashback in stable combustion, the fuel supply is cut-off shortly after flashback.

The pressure fluctuations were sampled from isothermal and reactive flows at sampling frequencies of $f_{s_{cold}} = 8192Hz$ and $f_{s_{hot}} = 16384Hz$ from the surface of the nozzle at a location upstream of the swirl vanes. Measurements were made for the flow velocities $U = 10ms^{-1}$, $U = 12ms^{-1}$, and $U = 14ms^{-1}$ (measured at the top edge of the bluff-body). The equivalence ratio was $\phi = 0.79$, $\phi = 0.73$, and $\phi = 0.74$ for the three flow velocity cases, respectively. Unlike the experiment of flashback in stable combustion, this experiment was carried out for three different cases of equivalence ratio and flow velocity. Hence, the following notation convention is adopted for making reference to the pressure time series from this experiment and the dynamical system that generated them. A pressure time series from the isothermal flow with flow velocity U is denoted as p_U and a pressure time series from the reactive flow is denoted as P_U. Thus, lower case p corresponds to the isothermal flow, upper case P corresponds to the reactive flow and the subscript indicates the flow velocity. Using this notation, the pressure time series are summarized in Table 3.1.

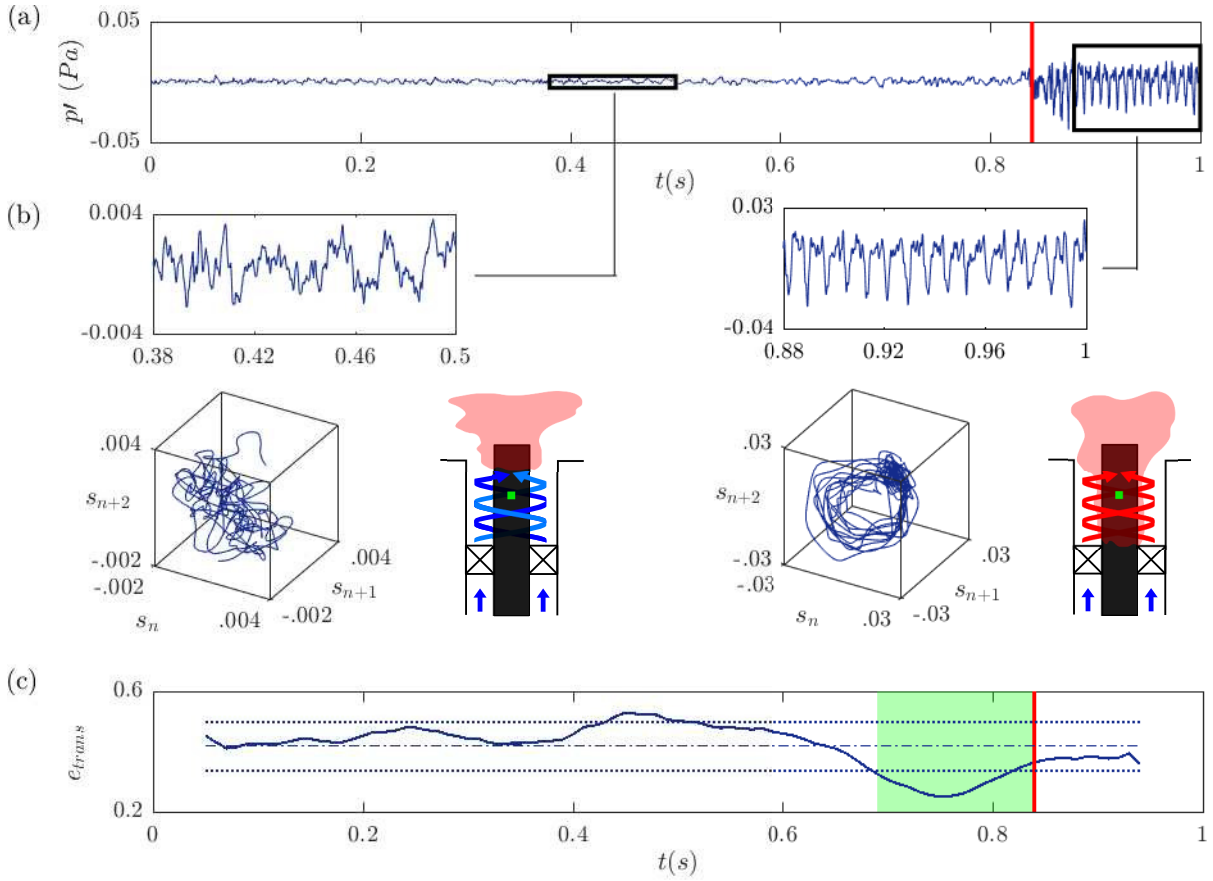

Figure 3.2: Pressure time series from the experiment of flashback in stable combustion: (a) — pressure time series, — onset of flashback, (b) time series sections before and after flashback with the corresponding embeddings in 3D phase space and schematics of the flame, and (c) — translation error, -·- mean, · · · standard deviation from mean, — onset of flashback.

3.3 Flame flashback in stable combustion

3.3.1 Precursors in translation error

The pressure time series from the first run of the flashback experiment in stable combustion is shown in figure 3.2a. The amplitude of the pressure fluctuations is small during the time that the flame is stable on the edge of the central bluff-body, where it is unconfined. When the swirl number is increased above a critical value the flame destabilizes and moves upstream into the nozzle. After the flame flashback the amplitude of the pressure fluctuations becomes large due to the coupling between the flame and the nozzle acoustics. Hence, the onset of flashback is approximately at the time that the amplitude of the pressure fluctuations starts to increase. In figure 3.2a, the approximate time of the flame flashback is shown with a (red) vertical rule at $t \approx 0.84s$.

Sections of the pressure time series in figure 3.2a from before and after flashback are shown in figure 3.2b with their corresponding embeddings in phase space and schematics of the flame's position. Prior to flashback the pressure fluctuations appear random and neighbouring sections

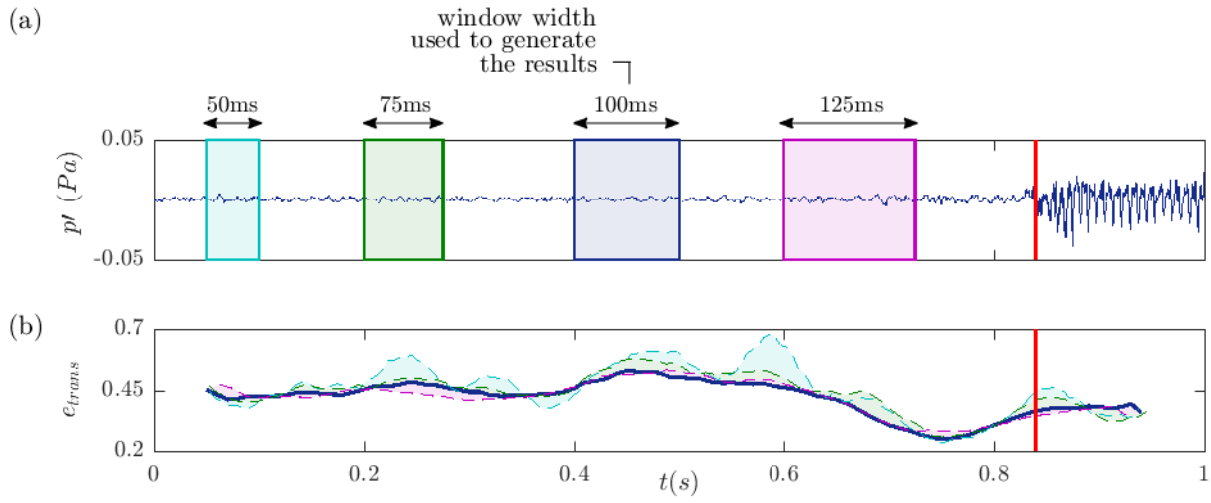

Figure 3.3: Window width independent solution: (a) the running windows of width 50*ms*, 75*ms*, 100*ms* (chosen width), and 125*ms* used to test for convergence of the translation error in the case of flame flashback in stable combustion and (b) the convergence of the translation error with increasing window width.

of the trajectory in phase space show a poor degree of parallelism. On, the other hand, after flashback, the pressure fluctuations appear somewhat periodic and neighbouring sections of the trajectory are showing a significantly better degree of parallelism. Application of the 0-1 test [64] to this part of the time series indicates that the dynamics during this interval are regular.

The translation error of the system's trajectory in phase space as a function of time is shown in figure 3.2c. The translation error clearly falls below the standard deviation from the mean just before flame flashback. This could be a precursor of the flame flashback observed in the experiment.

The translation error in figure 3.2c is calculated from the time series using a 100*ms* wide running window with 90% overlap and the parameters $M = 250, Q = 5, k = 3$. The time delay T and the embedding dimension D are chosen for each instance of the running window using the methods of average mutual information (AMI) and false nearest neighbours (FNN), respectively (see chapter 2). The chosen window width is based on a window width independence study that calculated the translation error using running windows of width 50*ms*, 75*ms*, 100*ms*, and 125*ms* and checked for convergence of the result. The windows used in the window width independence study are shown in figure 3.3a and the corresponding translation error calculations are shown in figure 3.3b. In figure 3.3b, the translation error is seen to converge for the chosen window width (100*ms*).

The translation error of the system in the four repetitions of the experiment and the ensemble average are shown in figure 3.4. The translation error of the system is calculated in all cases from the time series using the same parameters as those used for the first run of the experiment. That is, a 100*ms* wide running window with 90% overlap, the parameters $M = 250, Q = 5, k = 3$ and

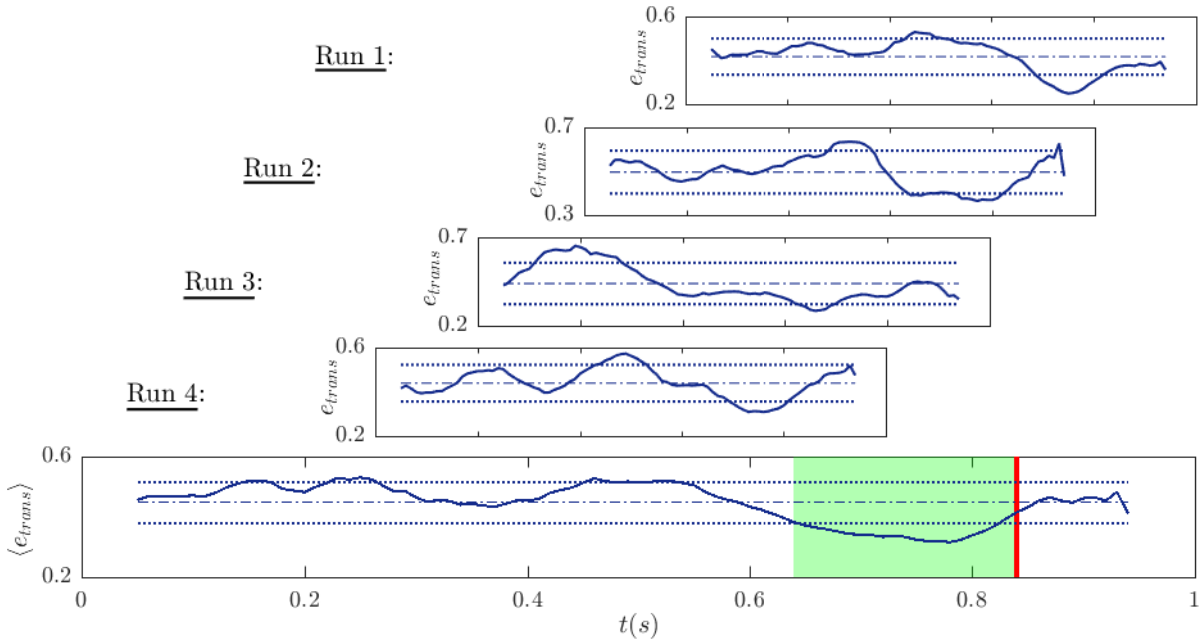

Figure 3.4: The translation error from the experiment runs and their ensemble average: -·- mean, ··· standard deviation from mean, — onset of flashback.

the time delay T and the embedding dimension D are chosen for each instance of the running window with the methods of AMI and FNN. The ensemble average translation error falls below the standard deviation from the mean just before flame flashback, confirming the observation in figure 3.2c from the first run of the experiment.

According to Miyano [129], a translation error less than 0.1 indicates regular dynamics, between 0.1 and 0.5 indicates either temporally correlated stochastic dynamics or regular dynamics contaminated with observational random noise, and values exceeding 0.5 indicate stochastic dynamics. The ensemble average of the translation error in figure 3.4 is in the range $0.45 - 0.5$ during the early stages of the experiment, then decreases to a minimum of 0.32 before the flame flashback and returns to its initial levels after the flame flashback. Therefore, according to the definition given by Miyano [129], the dynamics of this system are either regular and contaminated with observational random noise or stochastic with temporal correlations. Nonetheless, the ensemble average translation error decreases on approaching the flame flashback and thus, the determinism of the system dynamics increases temporarily just before the occurrence of the flame flashback. This is an important precursor of the flame flashback. A recurrence analysis of the data from the stable combustion case could not be performed because the pressure time series from this experiment are short and the recurrences are not enough to plot an accurate recurrence pattern for the system.

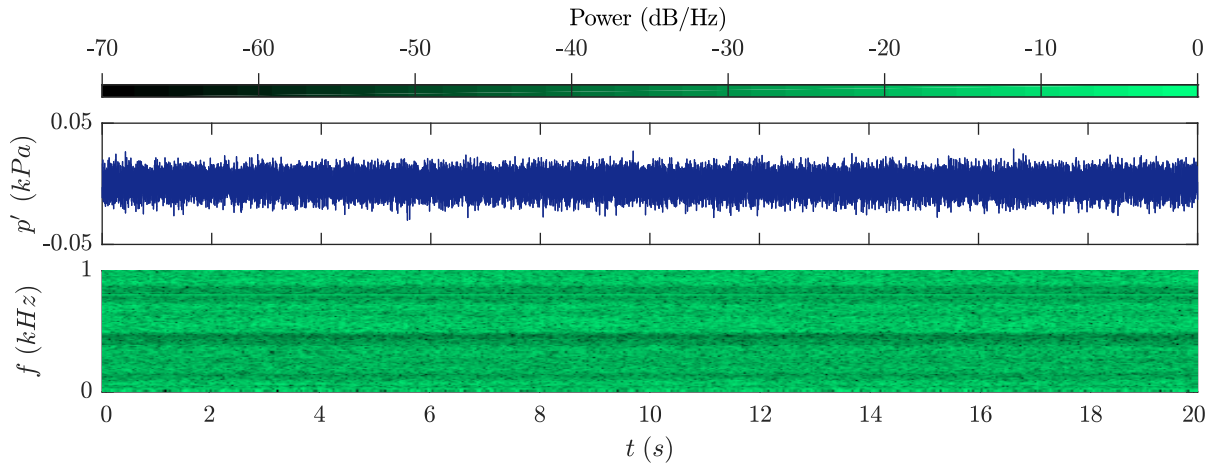

Figure 3.5: Pressure time series p_{10} from the isothermal flow and its spectrogram.

3.4 Flame flashback in unstable combustion

The flame in this experiment is in an enclosure and as such the flame interacts with the chamber acoustics. Hence, the frequency spectra of the pressure fluctuations are checked for precursors of the flame flashback of this experiment before NTSA of the pressure measurements.

3.4.1 Acoustic modes of the combustor

Pressure oscillations in the isothermal flow correspond to acoustics induced in the combustor by the turbulent flow. The pressure time series p_{10} from the isothermal flow is shown with its spectrogram in figure 3.5. The spectrogram shows that the pressure oscillations in the isothermal flow are broadband noise. The same observation is made for the higher flow velocity cases, p_{12} and p_{14}, which are not shown here.

For the reacting conditions, the pressure time series P_{10}, P_{12}, and P_{14} and their spectrograms are shown in figure 3.6. The amplitude of the pressure oscillations is about $1-2kPa$ before flashback. The jump in the amplitude of the pressure oscillations is a result of flame flashback. After flashback, the flame anchors in the upstream tube and the change results in high amplitude instability. The approximate time at which the amplitude of the pressure oscillations becomes large will be referred to hereafter as the approximate flashback point t_{flash}. The arrows in figure 3.6 indicate the approximate flashback point in the time series. It should be noted that the flashback event is an extremely fast process and it is, therefore, difficult to precisely determine the instant when it occurred in the pressure time trace. In the scale of the pressure traces in figure 3.6 the jump appears almost instantaneous and it is therefore convenient for the purpose of discussion to define the time instant of the jump as the nominal approximate flashback point. The approximate flashback points for the time series P_{10}, P_{12} and P_{14} are $t_{flash} = 6s$, $t_{flash} = 16s$, and $t_{flash} = 5s$ respectively.

The spectrogram of P_{10} in figure 3.6a shows that prior to flame flashback there are two

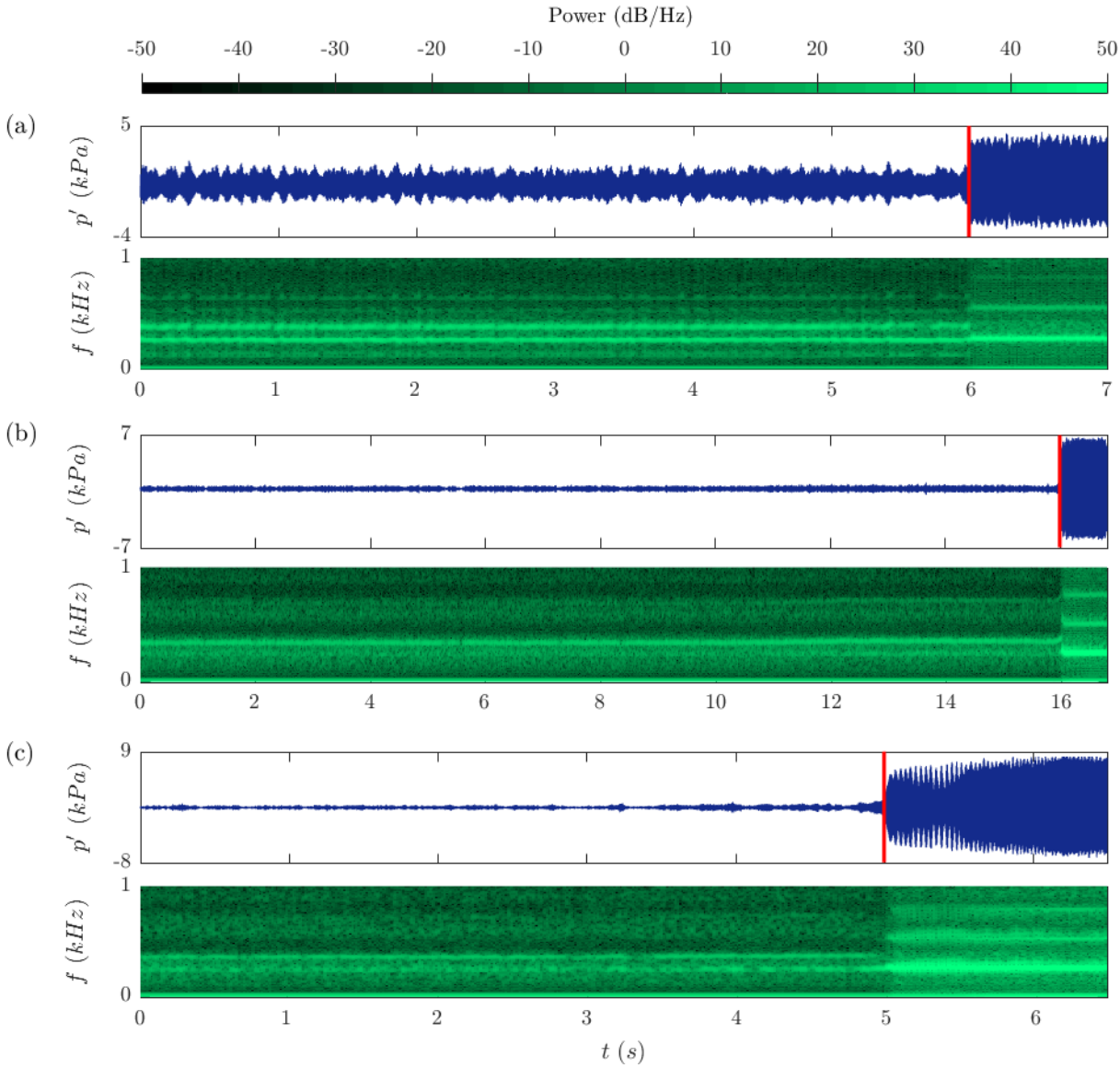

Figure 3.6: The pressure time series from the hot flows and their spectrograms: (a) P_{10}, (b) P_{12}, and (c) P_{14}. The (red) vertical rules indicate approximately the time instant at which flame flashback occurs.

dominant frequencies, at $270Hz$ and $390Hz$. The pressure oscillations at these frequencies have approximately the same power level. After flame flashback, again the pressure oscillates at two dominant frequencies, now at $270Hz$ and $540Hz$. The lower frequency after flame flashback ($270Hz$) is the same as the lower frequency before flame flashback, and the higher frequency ($540Hz$) is the first harmonic. The second harmonic can also be faintly seen. The spectrogram of P_{12} in figure 3.6b shows that there are two dominant frequencies before flame flashback, around $270Hz$ and $340Hz$. After flame flashback, there are three dominant frequencies for this case, at $270Hz$, $540Hz$, and $810Hz$. The latter two are the first and second harmonics. The spectrogram of P_{14} in figure 3.6c shows two dominant frequencies before flame flashback, at $270Hz$ and $360Hz$. After flame flashback, the dominant frequencies are at $270Hz$ and the first and second

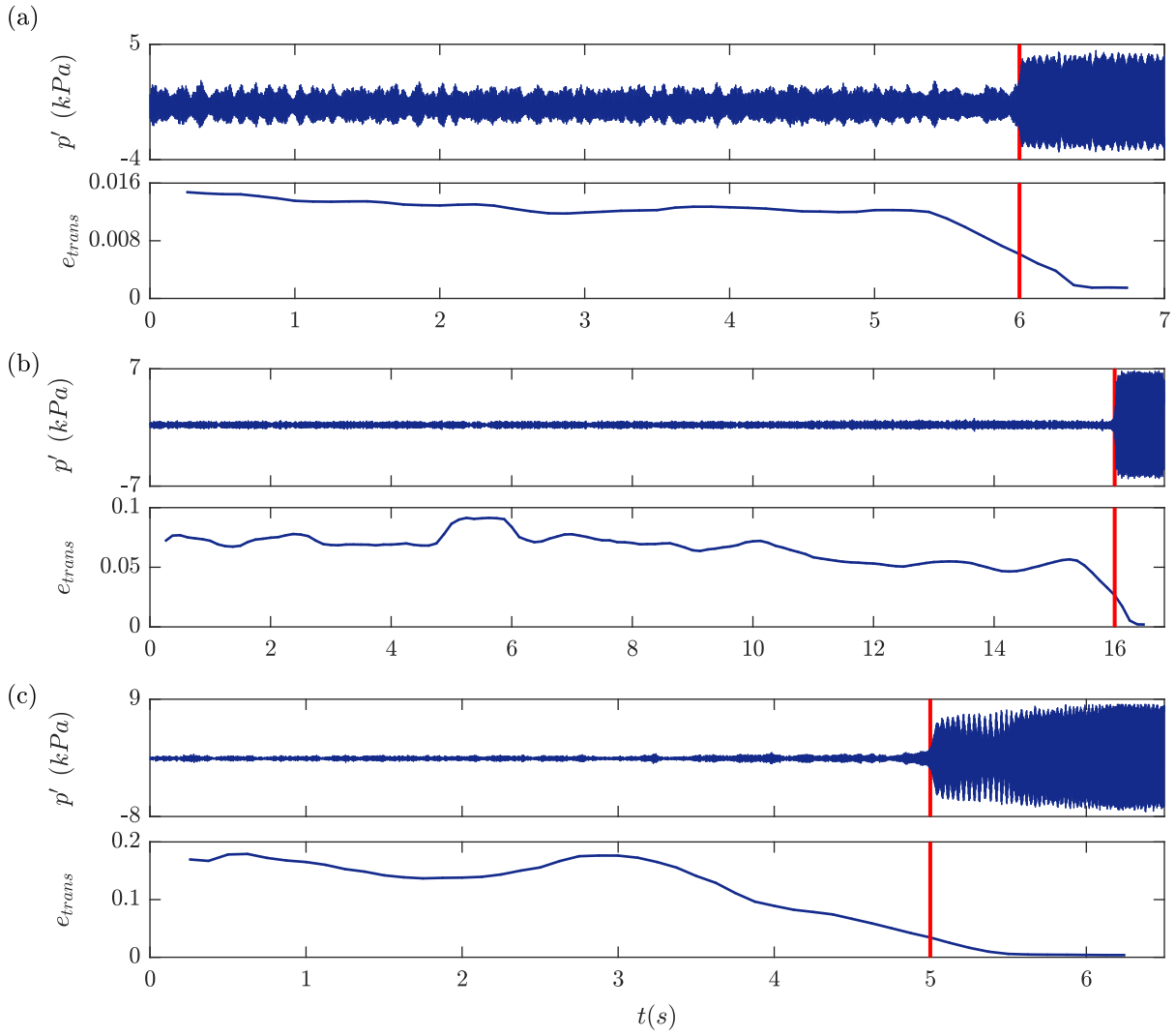

Figure 3.7: The pressure time series from the hot flows and the corresponding translation error of the reconstructed trajectory in phase space: (a) P_{10}, (b) P_{12}, and (c) P_{14}. The (red) vertical rules indicate approximately the time instant at which flame flashback occurs.

harmonic frequencies. In all cases, there is a sudden change in the dominant frequencies at the time of flame flashback. That is, there is no indication that the flame flashback is approaching.

3.4.2 Precursors in translation error

For the three cases of flame flashback in unstable combustion, the pressure time series and the translation errors of each system's trajectory in phase space as a function of time are shown in figure 3.7. In all cases, the translation error is calculated from the time series using a $100ms$ wide running window with 90% overlap and the parameters $M = 250, Q = 5, k = 3$. The time delay T and the embedding dimension D are chosen for each instance of the running window with the methods of AMI and FNN, respectively.

In figure 3.7a, the translation error is ≈ 0.012 long before flame flashback and begins to

decrease monotonically $\approx 0.62s$ before the flame flashback point and reaches the value of 0.006 at the moment of flame flashback. Similarly, in figure 3.7b, the translation error is ≈ 0.07 long before the flame flashback and begins to decrease monotonically $\approx 0.62s$ before the flame flashback point and reaches the value of 0.026 at the moment of flame flashback. In figure 3.7c, the translation error is ≈ 0.15 long before the flame flashback and begins to decrease monotonically $\approx 1.5s$ before the flame flashback point and reaches the value of 0.034 at the moment of flame flashback. The translation error decreases by 50%, 63%, and 77% just before flame flashback in these unstable combustion cases. The observation is in line with the observation made regarding the translation error of the system in the stable combustion case. Thus, the translation error decreases just before the flame flashback in both stable and unstable combustion cases. A reduction in translation error signifies that the determinism of the system increases before flame flashback. The time series from these experiments of flame flashback in unstable combustion capture a longer time span of the system behaviour compared to the time series from the stable combustion case. Therefore, the determinism of the system in the unstable combustion case can also be investigated through recurrence analysis.

3.4.3 Precursors in recurrence plots

The typical recurrence plot of the isothermal flows is shown in figure 3.8. It contains a homogeneous distribution of single isolated dots that resembles salt-and-pepper noise. A few short segments that can be observed are again a result of the acoustic resonance that is induced intermittently by the turbulent flow. This is in keeping with the spectrogram in figure 3.5. The features of the isothermal flow are in contrast with those of the reacting conditions that follow and are shown to elucidate how thermoacoustic coupling in the presence of the flame induces deterministic behaviour in the system.

The recurrence plot for the one second window of the P_{10} pressure time series from the reacting flow, two seconds prior to the approximate flashback point is shown in figure 3.9. The increased proportion of short line segments—deterministic structures—interrupted by isolated points or white horizontal and vertical spaces is clearly visible. Short line segments indicate a positive maximal Lyapunov exponent [44, 127] and hence, chaotic dynamics of the thermoacoustic coupling, that have also been identified in other studies [62, 80, 81]. A range of line lengths is required to describe the distribution of line segments in the RP, which suggests that a high level of complexity exists in the system dynamics. The vertical (or horizontal) white spaces indicate the presence of transients that occur roughly at intervals of 1-2 seconds. While the RP shown in figure 3.9 spans only one second of the time series, the RP for earlier sections of the pressure time trace are qualitatively similar. The observation also holds for the time series P_{12} and P_{14}.

The pattern formed by the collection of points and diagonally aligned line segments is a signature of the system dynamics long before flashback and is different for the other cases from

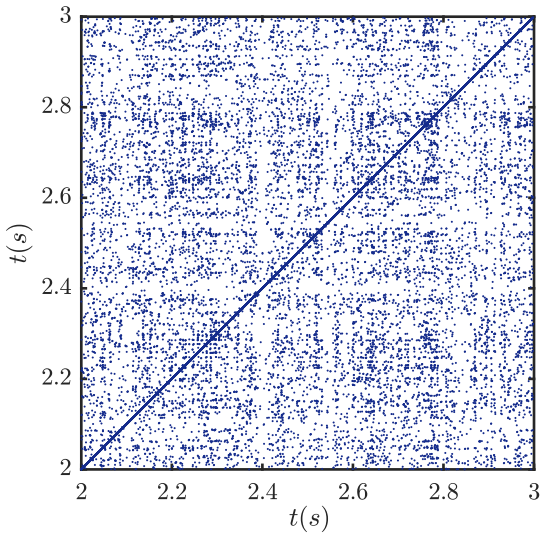

Figure 3.8: The recurrence plot constructed for a one second window of the pressure time series p_{10} from the cold flow using the parameters $T = 4, D = 5$ and $\varepsilon = 0.05 d_A$.

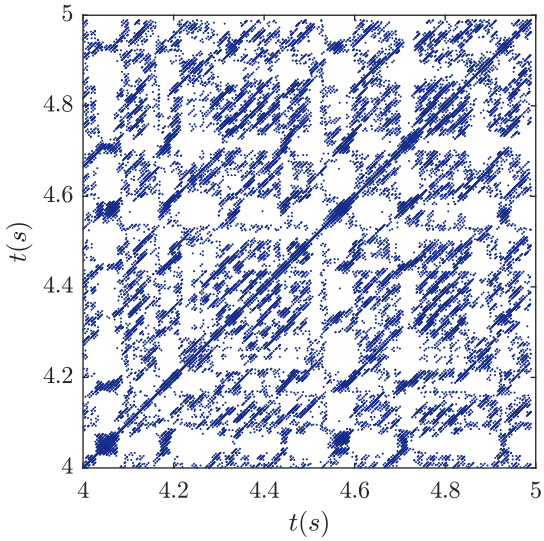

Figure 3.9: The recurrence plot constructed for a one second window prior to flashback for the case of P_{10} from the reactive flow using the parameters $T = 15, D = 4$ and $\varepsilon = 0.05 d_A$.

Table 3.2: The parameters used to construct the recurrence plots shown in figure 3.10.

Pressure time series	T $(data\ points)$	D (-)	ε $(\%d_A)$
P_{10}	15	4	5
P_{12}	14	6	4
P_{14}	13	6	4

which P_{12} and P_{14} are measured, primarily due to different flame dynamics and turbulence intensity of the base flow.

The RPs obtained for the one second windows of the time series P_{10}, P_{12} and P_{14} immediately prior to flashback (the right hand edge of the one second window coincides with the approximate flashback point of the time series) are shown in figure 3.10. The parameters that have been used to obtain these RPs are summarized in table 3.2.

It is immediately apparent that the RP in figure 3.10a, obtained from P_{10}, contains more dots that form diagonal lines than the RPs obtained from P_{12} and P_{14}. This implies, that immediately prior to flashback, there is more determinism in the dynamics of this system. The reason for this is believed to be the less turbulence in the base flow. At the larger Reynolds numbers associated with the higher flow velocity cases P_{12} and P_{14}, the fluid system is more turbulent (chaotic) and the flow properties are more sensitive to small disturbances. Hence, the state of the system is less deterministic and recurrences are short lived and less dots form diagonal lines in RPs.

The pattern seen in the RP in figure 3.10a is for a one second long window of the P_{10} time series just before the approximate flashback point. The pattern is notably different from that in RPs developed for earlier windows of P_{10} (see figure 3.9, for instance). Thus, the dynamics of the system are quite different just before flashback. The same is observed for P_{12} and P_{14}.

Further inspection of each individual recurrence plot in figure 3.10 reveals that there are changes of the pattern within the one second windows as one moves diagonally upwards and approaches the approximate flashback point (the right hand edge of each RP). To illustrate this, the RPs in figure 3.10 are divided into four sections by (red) horizontal and vertical rules. The pattern in the lower left sections of the RPs is different to the pattern in the upper right sections. Towards the end of the one second windows, a small region of a transience is identified (the horizontal and vertical white patches). This region is the largest for the case of P_{14}. This transience is due to the change in amplitude of the pressure as a result of flashback. Thus, flashback can be clearly identified in the RP. More interestingly, the RP also contains smaller changes even earlier. In the case of P_{10}, the solid (red) vertical and horizontal rules at $\approx 5.42s$ in figure 3.10a separate two regions along the main diagonal that are visually distinguishable. The same is inferred for the RPs of cases P_{12} and P_{14}.

These changes observed in the structure of the recurrence plots immediately prior to flame flashback could be an indication that flashback is about to occur. That is, flashback could have precursors associated with subtle dynamical changes in the system.

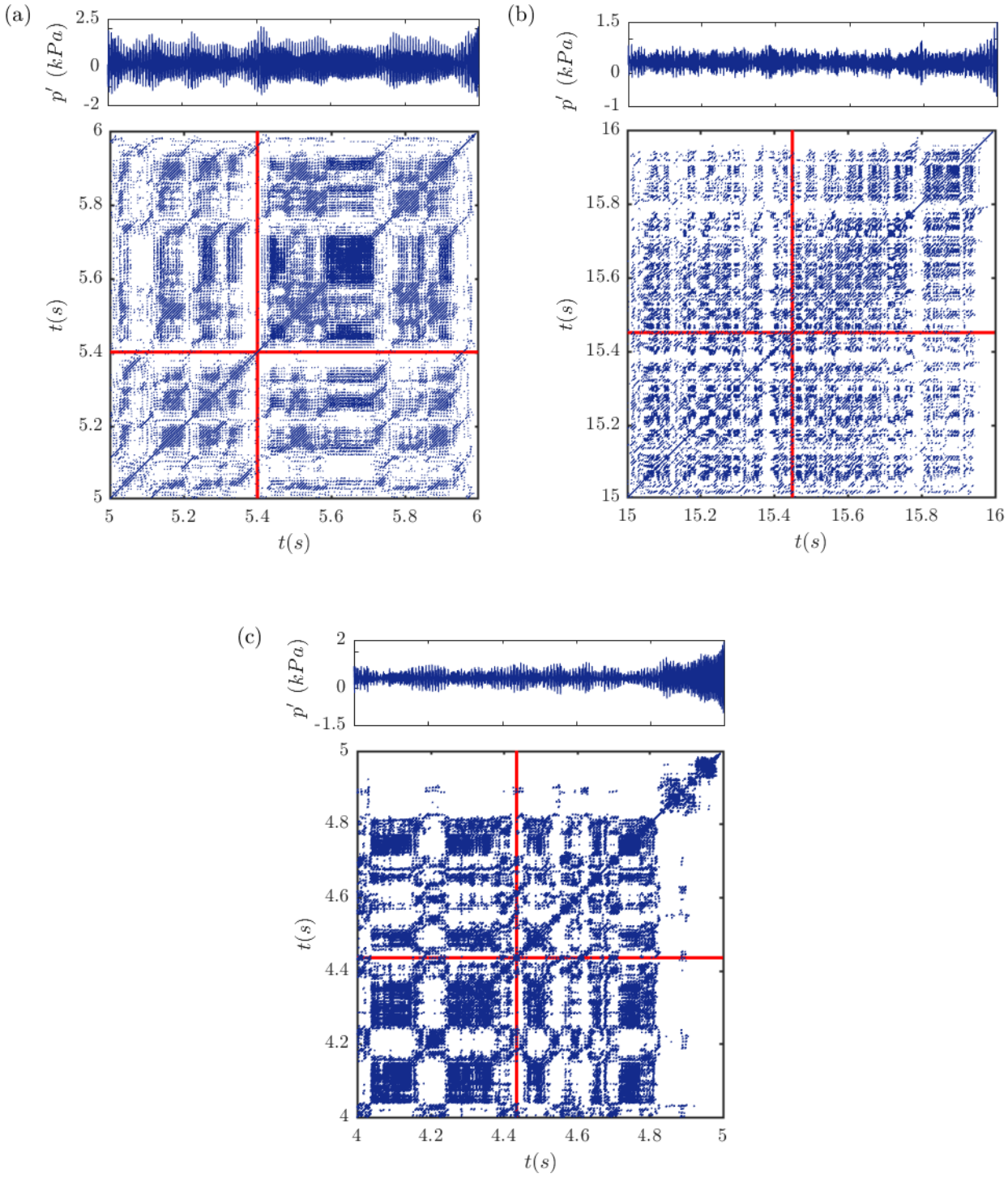

Figure 3.10: Recurrence plots constructed for the one second window immediately prior to flashback: (a) P_{10}, (b) P_{12}, and (c) P_{14}. The solid (red) rules separate the RP along the main diagonal into two parts with visually differing texture.

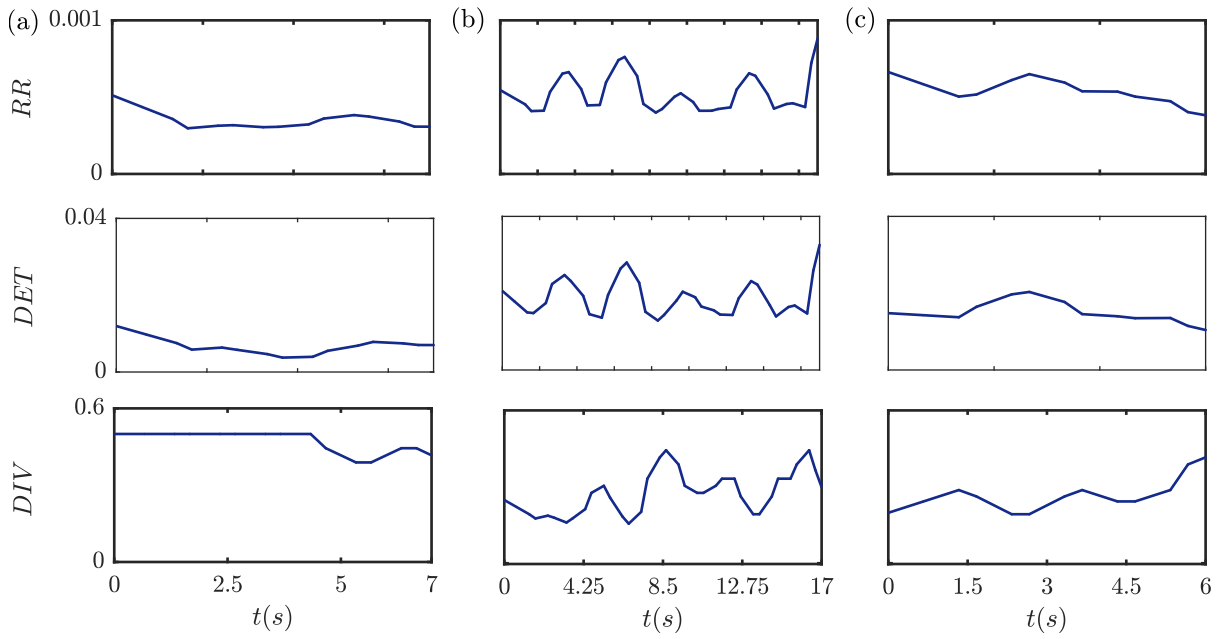

Figure 3.11: The recurrence rate (*RR*), determinism (*DET*), and divergence (*DIV*) for the isothermal flows for a one second running window with 90% overlap: (a) p_{10}, (b) p_{12}, and (c) p_{14}.

3.4.4 Precursors in RQA measures

Further support to this observation is obtained from recurrence quantification analysis (RQA) [173, 127, 44]. The recurrence rate (*RR*), determinism (*DET*), and divergence (*DIV*) as a function of time are obtained using a one second wide running window with 50% overlap and are shown in figure 3.11 for the isothermal flow and in figure 3.12 for the reactive flow. Despite using overlap, the time traces of the RQA measures required a smoothing operation. A moving average gives the smooth time traces shown in figures 3.11 and 3.12.

The *RR* in figure 3.11a-c for the case of the isothermal flows is low and does not change considerably over time. The magnitude of RR is associated with the choice of recurrence threshold ε. For a large ε multiple recurrences of the trajectory in phase space fall into the neighbourhood being searched and are counted as a single recurrence. Conversely, for a smaller ε, less recurrences fall into the neighbourhood and hence, more recurrences are counted. In table 3.2, the recurrence threshold used in the case of P_{12} and P_{14} is smaller than that used in the case of P_{10}. Thus, the *RR* in the case of P_{10} in figure 3.12a is smaller than in the case of P_{12} and P_{14} in figures 3.12b and 3.12c. As the isothermal flows are dynamically similar, the *RR* curves for the three cases are statistically stationary.

The *DET*, which can take values between 0-1 inclusive, is very low for the isothermal flows in figure 3.11, thus, indicating a highly stochastic process. This is expected in accordance with previous discussions of the isothermal flow. Similarly to the *RR* curves, the *DET* changes are statistically stationary, thus indicating that there are no significant trends present in the system dynamics.

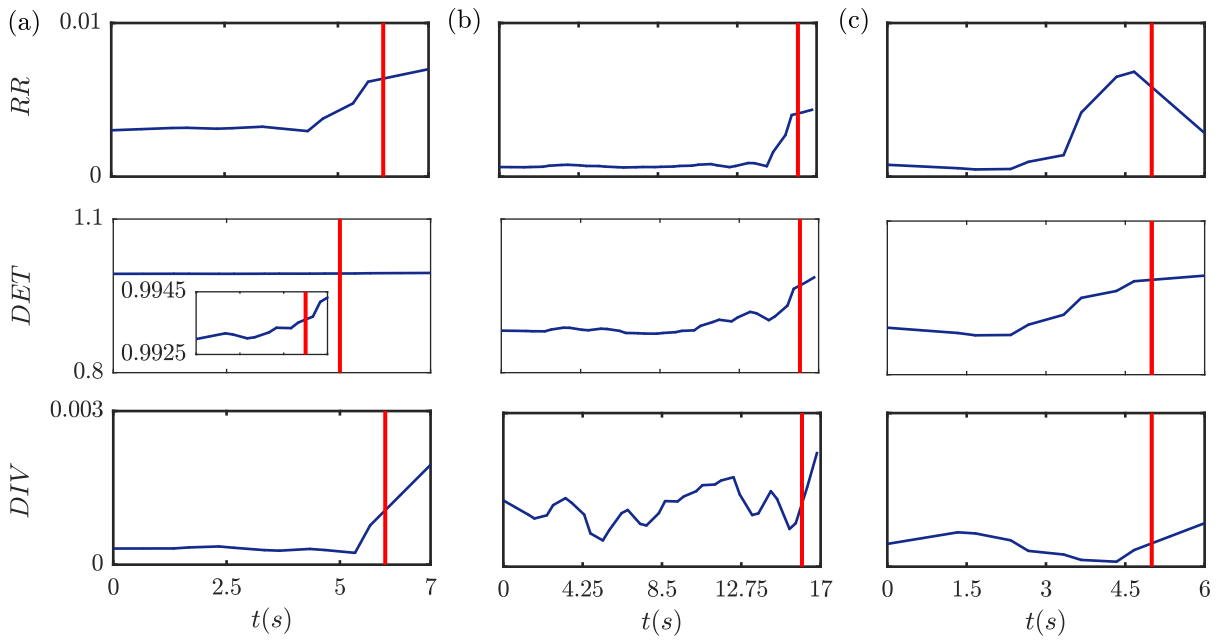

Figure 3.12: The recurrence rate (*RR*), determinism (*DET*), and divergence (*DIV*) of the pressure time series from the reactive flows calculated using a one second wide running window with 50% overlap: (a) P_{10}, (b) P_{12}, (c) P_{14}. The (red) vertical rules indicate t_{flash}.

The *DIV* stays close to 0.5 for the isothermal flows, which corresponds to the prescribed threshold for the minimum line length $l_{min} = 2$. The *DIV* shows that the isothermal cases are dynamically similar and that the RPs are comprised primarily of isolated points.

The *DET* in figure 3.12 for the reacting flows is close to unity. This is in accordance with the visual interpretation of the RPs in figure 3.9 and figure 3.10, which consist of distinctly visible diagonal lines. The reacting flow cases are clearly more deterministic when compared to the isothermal flow due to the existence of the thermoacoustic coupling. More importantly, the *DET* curves for all three cases, P_{10}, P_{12}, and P_{14} show a distinct trend prior to flashback, that is the *DET* is not statistically stationary. The determinism begins to increase towards the onset of flashback and reaches a maximum after flashback. The (red) rules in figure 3.12 indicate the approximate flashback point. In the case of P_{10}, the increase of the *DET* towards the onset of flashback occurs on a much smaller scale than in the other cases and the increase is shown in the inset plot in figure 3.12a.

The *RR* for the reactive flows is around an order of magnitude larger than that of the isothermal flows, indicating that there are significantly more recurrences occurring in the reactive flow. The trend of the *RR* is similar to that of the *DET*. That is, it increases prior to the onset of flashback. The increase starts two to four seconds prior to flashback.

The *DIV* curves are more difficult to generalize for the three cases from the reactive flow. In the case of P_{10}, there is an obvious jump prior to the onset of flashback. On the other hand, in the cases of P_{12} and P_{14} there appears to be a gradual increase of the *DIV* as flashback is approached, however the increase is extremely slow. 1.8914 9.7589 13.8834 1.6000

For a thermoacoustically unstable system undergoing flashback, the results show that subtle temporal changes occur in the dynamics of the coupling prior to flashback. These changes are made apparent through the characterization of the recurrences of phase space trajectories reconstructed from pressure time traces, in a multi-dimensional phase space.

The indication of precursor behaviour is first found in the RPs, where the patterns in the RPs change as flashback is approached. A change in the distribution of deterministic structures (lines) and non-deterministic structures (isolated points) leads to changes in the patterns found in the RPs. Furthermore, through RQA, it is found that before flashback the determinism in the system shows an increasing trend as an increasing number of isolated black dots (isolated recurrences) come together to form diagonal lines (consecutive recurrences). The recurrence rate also increases simultaneously; this could be related to the fact that before flashback, there is a relative increase in the sections of trajectories recurring and hence, in the overall number of recurrences at the same ε. In the transition from an asymptotic state before flashback to another attractor after flashback, while the actual flashback event might be an instantaneous process, changes that occur in system dynamics are not.

These trends that have been identified show that as flashback is approached, the dynamics of the flame and hence, the flame-acoustic coupling changes. The duration of relevant changes that occur prior to flashback span two to four seconds in the cases considered in the current work. These observations lead to two important implications. Firstly, that precursors of flashback exist, and with appropriate methods, it is possible to design flashback detection and control systems. Secondly, the phenomenon of flashback is a result of interactions that span a duration of seconds. In previous studies on the mechanisms of flashback [98, 155, 69], discussions focus on the stages involved in the evolution of the flame and hydrodynamic structures during a flashback event. It is very likely that observable qualitative changes that have been associated with the occurrence of flashback are also related to changes in the recurrence measures obtained from the pressure time traces in the current work.

Although flashback in a thermoacoustically unstable combustion is analysed, it is expected that a similar analysis of other flashback mechanisms should yield similar results on flashback precursors. Recurrence analysis, which enables quantification of system complexity and determinism, appears to be an appropriate method to identify transitional changes/precursor behaviour present in the extremely complex dynamics that are involved in the process of flame flashback.

3.5 Conclusions

The current work has found that subtle changes occur in the system dynamics prior to flame flashback in stable and unstable combustion. Linear analyses in the time and frequency domains fail to detect these changes. However, analysis of the trajectory of the system in phase space,

which has been reconstructed from pressure time series measurements obtained from the system, has successfully detected flashback precursors.

The phase space trajectory of the studied systems has been analysed through its translation error and recurrence characteristics. The translation error trajectories begins to decrease as flashback is approached, which indicates that neighbouring sections of the trajectory start to show a larger degree of parallelism. That is, the trajectory in phase space starts to becomes more organised. Hence, the determinism of the system increases as the flashback point is approached. In the recurrence plots, as the flashback point is approached, the number of recurrences increases and more of them start to form diagonal lines and the visual patterns in the recurrence plots start to change. This observation is put on a firm footing by the recurrence quantification measures of RR and DET, which measure the recurrence density and recurrences forming diagonal lines respectively. The RR and DET both increase prior to the onset of flashback. In conclusion, it can be said that flame flashback is induced as a result of a specific interaction among flame, acoustics, and hydrodynamics, which causes an increase in the determinism of the system dynamics.

The findings have important academic as well as industrial implications. Future investigations on flashback mechanisms could benefit from the insight that relevant dynamical changes occur in the flame before the flashback. Furthermore, the results demonstrate the feasibility of advanced detection approaches that could be used in combustion control systems to actively suppress flame flashback. Since, the determinism of the system begins to increase at least one second before flashback and the calculation of the translation error for a window of the time series takes only one tenth of a second, the method could potentially be used to detect the flashback in real time. The use of the RQA measures for real time detection is not as readily realisable because their calculation for a window of the time series takes a few minutes. Nonetheless, significant time savings from implementing the algorithms in Python or C, which are usually used in control applications, and from combining these detection methods with short-term prediction of the future path of the trajectories could make the use of recurrence analysis for active control more feasible. An alternative approach for the active control of flame flashback could be its forecast using machine learning algorithms instead of its early detection using the approaches of the current work. However, in comparison, the machine learning approach has many disadvantages including the need for a large amount of data covering many test cases and the absence of physics. Although there is a proliferation of machine learning algorithms that are extremely efficient, these require a large number of training data in order to operate successfully and for a highly stochastic phenomenon such as flame flashback the data that will be required to train these algorithms to predict flashback is anticipated to be very large indeed. Furthermore, large amounts of data are not easily obtainable for flame flashback because data collection relies heavily on experiments, which are difficult to conduct and require expensive diagnostic equipment. This is because simulations of flame flashback are near to impossible given that the phenomenon is extremely transient and involves a multitude of physics. Regardless of the approach taken to

forecast flame flashback, it is important to note that the actions taken to suppress it could in effect trigger the reverse instability, flame blow-off.

Chapter 4

Modelling of an entropy wave advecting in a turbulent channel flow

4.1 Introduction

The unsteady heat release in a fluid flow is a source of non-isentropic temperature oscillations widely referred to as entropy waves [37, 40, 86, 87]. In gas turbine combustors, these temperature perturbations incurred at the flame advect through the combustor and reach the convergent exit nozzle where their acceleration results in the generation of acoustic waves [174, 73]. The sound generated by the acceleration of entropy waves is accordingly termed entropy noise or indirect combustion noise, which in comparison with direct combustion noise is far less understood [19, 131]. Generation of entropy noise is subject to the survival of entropy waves throughout the advection process in the combustor. Yet, the extent of such survival continues to be a matter of contention [45, 46, 132].

Thermoacoustic stability is a crucial consideration in the design and development of premixed combustors [115, 143]. Hence, the acoustic response of subsonic and supersonic nozzles to entropic forcing, which, could influence thermoacoustic stability, has received significant attention [122, 130, 59, 113, 42, 74]. A commonplace assumption in all these investigations is that the amplitude of the entropy wave that impinges the nozzle inlet is spatially uniform and that the entropy wave can be approximated as a one-dimensional convective wave. However, there is now an emerging body of evidence indicating that advection of entropy waves can include sophisticated spatio-temporal dynamics, which can readily violate the one-dimensional assumption [17, 52]. Low order modelling of such dynamics remains as an ongoing challenge.

The early dispersion model of Sattelmayer [151] is the first to somewhat account for the influence of combustion chamber aerodynamics on an advecting entropy perturbation. The modelling approach treats an experimental dual fuel burner as a single-input single-output (SISO) dynamical system. The impulse response of the system is taken as the probability density function (PDF) of the residence time and is modelled by a rectangular pulse to yield an analytic

expression for the system transfer function. The transfer function is subsequently used in a thermoacoustic model to show that entropy waves do not have a destabilizing effect. The same conclusion is reached when the dispersion model is used in the low-order modelling of Eckstein and Sattelmayer [45].

Goh and Morgans [61] added a dissipation factor to the dispersion model of Sattelmayer [151] to account for the effect of entropy sinks and incorporated it into a thermoacoustic model other than that used by Sattelmayer [151]. Subsequently, the authors showed through a series of case studies that entropy noise could act constructively or destructively on combustor stability depending on the levels of dissipation and dispersion. Morgans et al. [132] then investigated the dissipation and shear dispersion of a passive entropy perturbation by performing an incompressible, direct numerical simulation (DNS) of an entropy perturbation in a fully-developed, turbulent channel flow. Dissipation of the entropy wave was defined in terms of the total thermal energy and as would be expected for an adiabatic system, was found negligible. Shear dispersion was modelled using the Sattelmayer [151] model but in this case it was shown that the response is better captured by a Gaussian pulse. Finally, a simple case study using the modified dispersion model and conditions representative of a typical gas turbine combustor revealed that the magnitude of the transfer function is significant up to frequencies relevant to combustion instabilities (several hundred Hertz). Further, numerical and experimental evidence also suggests that entropy noise can greatly influence combustion stability [8, 9, 70, 71, 134]. However, there also exist counter-arguments asserting that entropy waves are diffusive and thus, do not affect the thermoacoustic stability of combustors [151, 45, 46].

Recently, Giusti et al. [58] modelled the magnitude of the transfer function directly (instead of the system response) with an exponential function and showed that it scales well with a local Helmholtz number based on the entropy wavelength and streamwise distance. Although not realistic for a real combustor, the entropic forcing used was a single frequency sinusoid, and thus, any effects due to modal coupling were not included. Somewhat similarly, Wassmer et al. [171] fitted the analytic solution of the energy equation for a convection-diffusion process to the experimental measurements. An effective diffusivity included the transport effects of the turbulence.

In keeping with the linear one-dimensional framework of nozzle response studies, the models developed for the decay of entropy waves in the combustor [151, 132, 58] can readily be integrated into the existing thermoacoustic models. However, the linearity of the dynamics in these models is not based on simplifying assumptions with respect to the underlying physical processes responsible for the decay of the entropy perturbations, but is simply phenomenological. This is because the mechanisms responsible for the attenuation of the waves are largely unexplored [52]. Furthermore, the opposing findings with respect to the influence of entropy noise on combustion stability implies that turbulent mixing downstream of the flame plays an important role in the attenuation process that is not clearly understood yet. Thus, a more generic

modelling approach that allows for a small degree of non-linearity is warranted. Furthermore, the one-dimensional treatment of the entropy perturbation restricts the applicability of the existing models to SISO (single input, single output) thermoacoustic models [114, 138]. However, experimental [17] and numerical [52] studies have shown that entropy perturbations could become spatially uncorrelated. Hence, application of a cross-sectional average effectively filters out any spatially uncorrelated frequency components of entropy perturbations. The threshold frequency at which correlation starts to breakdown, and therefore, the amount of thermal energy that is filtered out by application of an average depends on the thermal boundary conditions and the hydrodynamics of the flow [52]. Consequently, it is imperative to model entropy waves as a multi-input-multi-output (MIMO) system. This, in turn, calls for prediction of the spatio-temporal dynamics of these waves. Yet, currently there is no low order modelling tool for this purpose. To address this issue, a low order model (LOM) that can simulate the amplitude decay and spatial distortion of a two-dimensional entropy wave is developed using a novel dynamical approach. The proposed LOM is capable of predicting both spatial and temporal features of an entropy wave. This is on the basis of reduction of the data generated by a DNS of compressible, fully-developed, turbulent channel flow with adiabatic and heat transferring external walls and an added Gaussian entropy perturbation. The DNS uses the same Reynolds number and shape of entropy perturbation as that of Morgans et al. [132]. However, unlike Morgans et al. [132], the current DNS is of a compressible flow, where the energy equation is not treated as a passive scalar equation but is solved simultaneously with the continuity and momentum equations. Furthermore, the important case of heat transferring walls is also simulated in addition to the adiabatic walls case considered in Morgans et al. [132].

4.2 High order modelling

A compressible, fully developed, turbulent airflow in a channel with an added entropy perturbation is simulated for adiabatic and non-adiabatic boundary conditions at the walls.

4.2.1 Computational model

A schematic of the channel configuration is shown in figure 4.1. The friction Reynolds number of the fully developed flow is $Re_\tau = 180$, which corresponds to Reynolds number $Re_c = 3367$ and Mach number $M_c = 0.15$ based on mean centreline velocity. The size of the simulation domain is $4\pi\delta \times 2\delta \times \pi\delta$ in the streamwise (x), wall-normal (y), and spanwise (z) directions respectively, where δ is the channel half-height. The size of the grid is accordingly $368 \times 128 \times 128$ nodes. Periodic boundary conditions are imposed in the streamwise and spanwise directions and the non-slip boundary condition at the walls. In the periodic directions the grid is uniform with $\Delta x^+ = 6.1$ and $\Delta z^+ = 4.4$. In the wall-normal direction the grid is stretched from $\Delta y_w^+ = 0.15$ at the wall to $\Delta y_c^+ = 5$ at the centreline according to $\Delta y_{j+1}/\Delta y_j = SF$, where $j = 1, 2, \ldots, 63$ is the

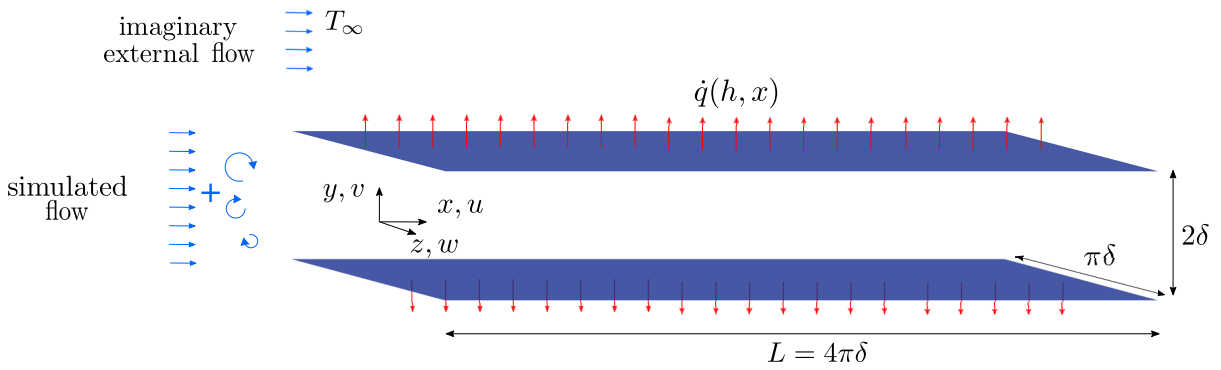

Figure 4.1: Channel configuration

cell number ($j = 1$ is the cell at the wall and $j = 63$ is the cell at the centreline) and $SF = 1.05$ is the stretch factor. The convective heat loss at the walls is controlled by setting the value of the heat transfer coefficient $h[Wm^{-2}K^{-1}]$. The temperature of the imaginary external flow is $T_\infty = 300K$. In the adiabatic case, the heat transfer coefficient at the walls is $h = 0Wm^{-2}K^{-1}$. The mean temperature of the flow in this case is $\overline{T} = 1500K$. The properties of air at this temperature are taken as $\rho = 0.2322kgm^{-3}$ and $\mu = 5 \times 10^{-5}kgm^{-1}s^{-1}$. In the non-adiabatic case, the heat transfer coefficient at the walls is $h = 200Wm^{-2}K^{-1}$. The mean temperature of the flow in this case is a function of the spatial coordinates and time.

4.2.2 Validation of turbulent flow

The cross-sectional profiles of the mean velocity and mean square of the turbulent fluctuations are shown in figure 4.2. Mean and fluctuating velocity components are indicated by an overbar ($\bar{}$) and prime ($'$) respectively. Velocity is non-dimensionalized by the wall-shear velocity $u_\tau = \sqrt{\tau_w/\rho}$, where τ_w is the flow shear stress on the wall. The wall-normal coordinate $y^+ = (u_\tau y)/\nu$, where ν is kinematic viscosity is the distance from the wall in wall-units. The profiles collapse with the canonical data of Kim et al. [96], thus, confirming that the grid is sufficiently fine and that the fully developed flow condition has been reached. The spatial variability of the velocity is elucidated by a snapshot of the instantaneous velocity field (velocity is non-dimensionalized by the local speed of sound) shown in figure 4.3 for a streamwise cross-section at midspan.

4.2.3 Simulated entropy wave

Snapshots of the convecting entropy wave after it has been added to the flow are shown in figure 4.4 for the case of adiabatic walls and in figure 4.5 for the case of convective heat loss at the walls. The amplitude of the wave is $\Delta T/\overline{T} = (T - \overline{T})/\overline{T}$, which is scaled using the mean flow temperature $\overline{T}$. In the adiabatic case, $\overline{T}$ is constant. However, it is important to note that in the non-adiabatic case $\overline{T}$ is a depends on the spatial coordinates and on the time. In both cases, the snapshot in the first row shows the state of the entropy wave as soon as it has been added

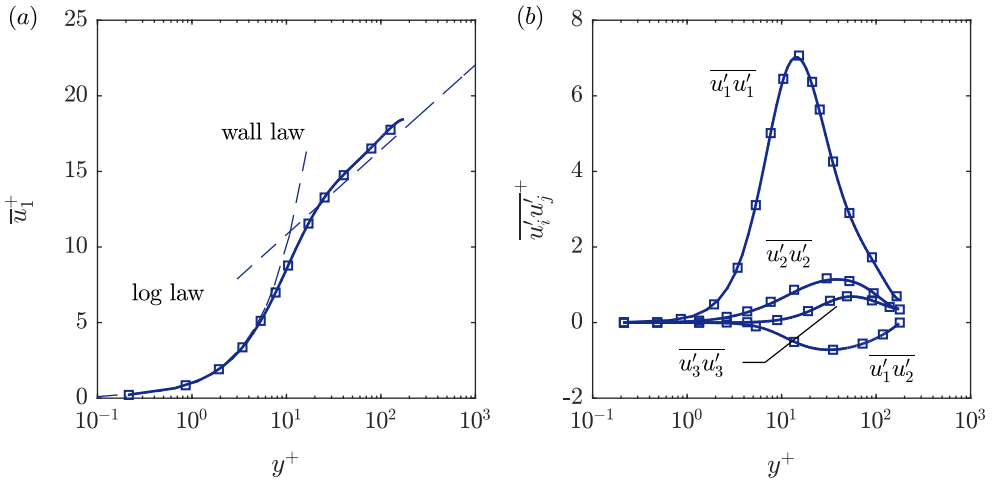

Figure 4.2: The flow (a) mean velocity and (b) mean square of turbulent fluctuations collapse with the canonical data of Kim et al. [96]: —— present study, □ Kim et al. [96]

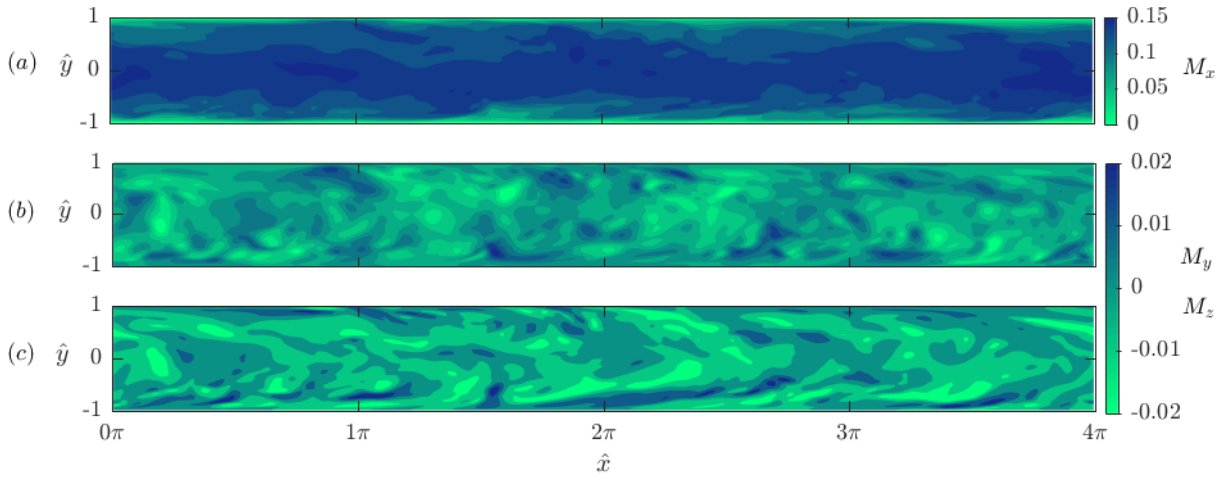

Figure 4.3: Instantaneous velocity component in the (a) streamwise, (b) wall-normal and (c) spanwise directions non-dimensionalized by the speed of sound

to the flow i.e. when the forcing of temperature at the insert plane has ceased (in figure 2.12 this corresponds to $\tau \approx 0.07$). The snapshots in the second and third rows show the state of the entropy wave when it reaches the channel half-length and outlet respectively. The time difference between the snapshots is the same and thus, the wave speed appears to be constant.

The maximum amplitude as soon as the entropy wave is added (in row one of figures 4.4 and 4.5) is $\Delta T/\overline{T} = 0.06$, which, is down from the peak value $A = 0.1$ of the added Gaussian perturbation (see equation 2.31). The 40% reduction in amplitude occurs in the time taken to generate the entropy wave, that is, in a non-dimensional time interval $\Delta\tau = 0.07$ (see figure 2.12a). By the time the wave reaches the channel half-length (in row two of figures 4.4 and 4.5) and outlet (in row three of figures 4.4 and 4.5), the maximum amplitude is $\Delta T/\overline{T} \approx 0.025$ and $\Delta T/\overline{T} \approx 0.015$ respectively. Thus, a 40% reduction in amplitude within $\Delta\tau = 0.25$ during the generation of the wave is followed by further 60% and 40% reductions within successive $\Delta\tau = 0.5$ intervals. It

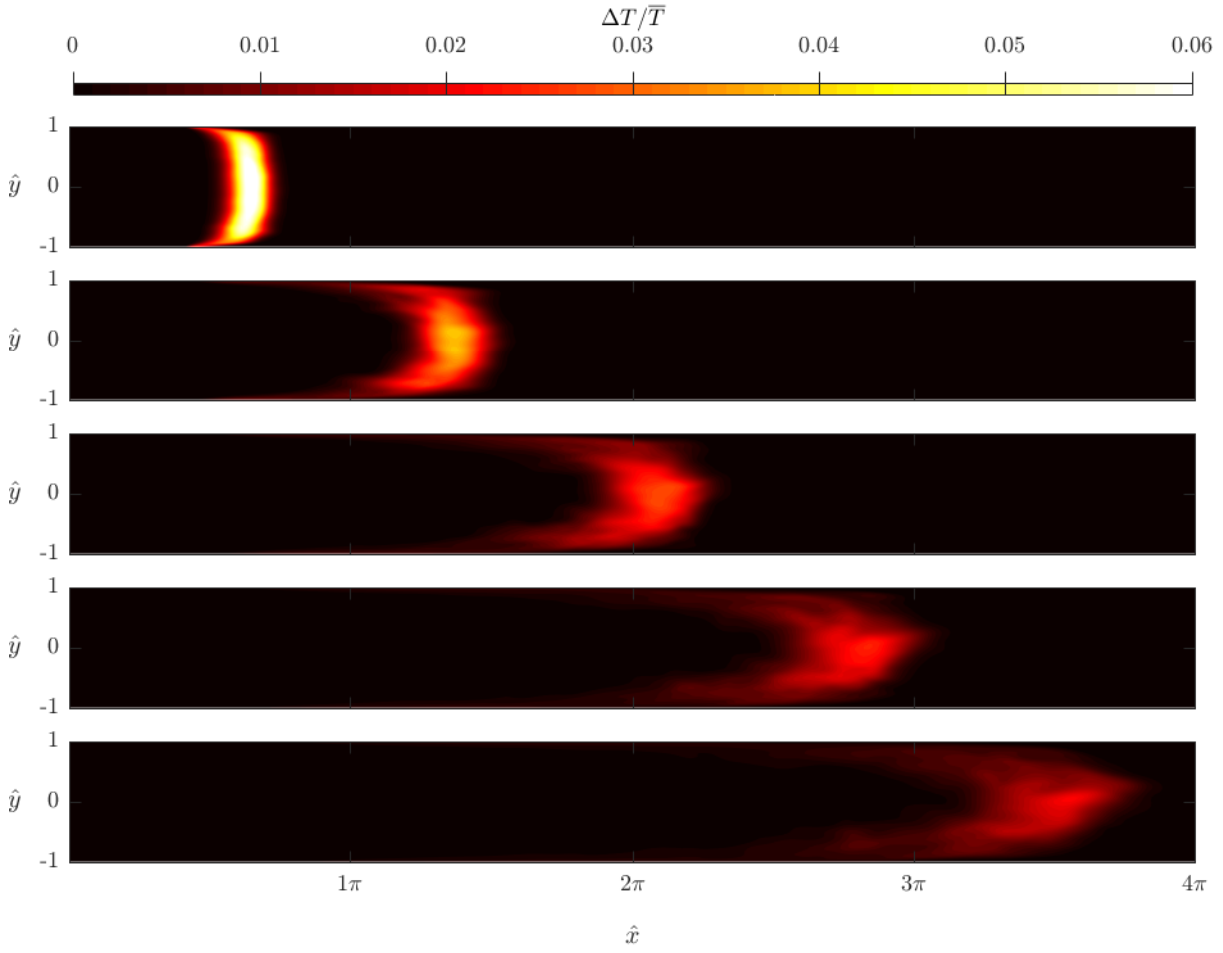

Figure 4.4: Convection of temperature perturbation through a fully developed, turbulent, channel flow with adiabatic walls obtained by direct numerical simulation of the high order system.

is clear that profound decay occurs while the entropy wave is being generated. The reason for this sharp drop in wave strength at the generation stage is twofold. First, the adverse temperature gradient that exists near the insert plane presents a strong driving force for heat transfer by molecular diffusion. In addition, turbulent mixing is also more effective than at later stages because the thermal energy that is effectively being added by perturbing the temperature field is distributed over a broadband frequency range (the width of the frequency spectrum depends on the value used for σ in equation 2.31 — the smaller σ is the wider the frequency spectrum). The thermal energy contained in the high frequency components of the perturbation that have small wavelengths of order comparable or smaller than that of the turbulence undergo strong turbulent diffusion. The extend of amplitude decay during the generation of the entropy wave is dependent on the temporal 'sharpness' of the added perturbation, which, in the case of the artificially generated wave in the present study is controlled by the value of σ in equation 2.31.

In both cases (figures 4.4 and 4.5), as the entropy wave convects from the insert plane to the channel outlet it undergoes changes in shape, thickness and amplitude. It will become clear from the discussion that follows regarding the shape and thickness of the wave that amplitude decay

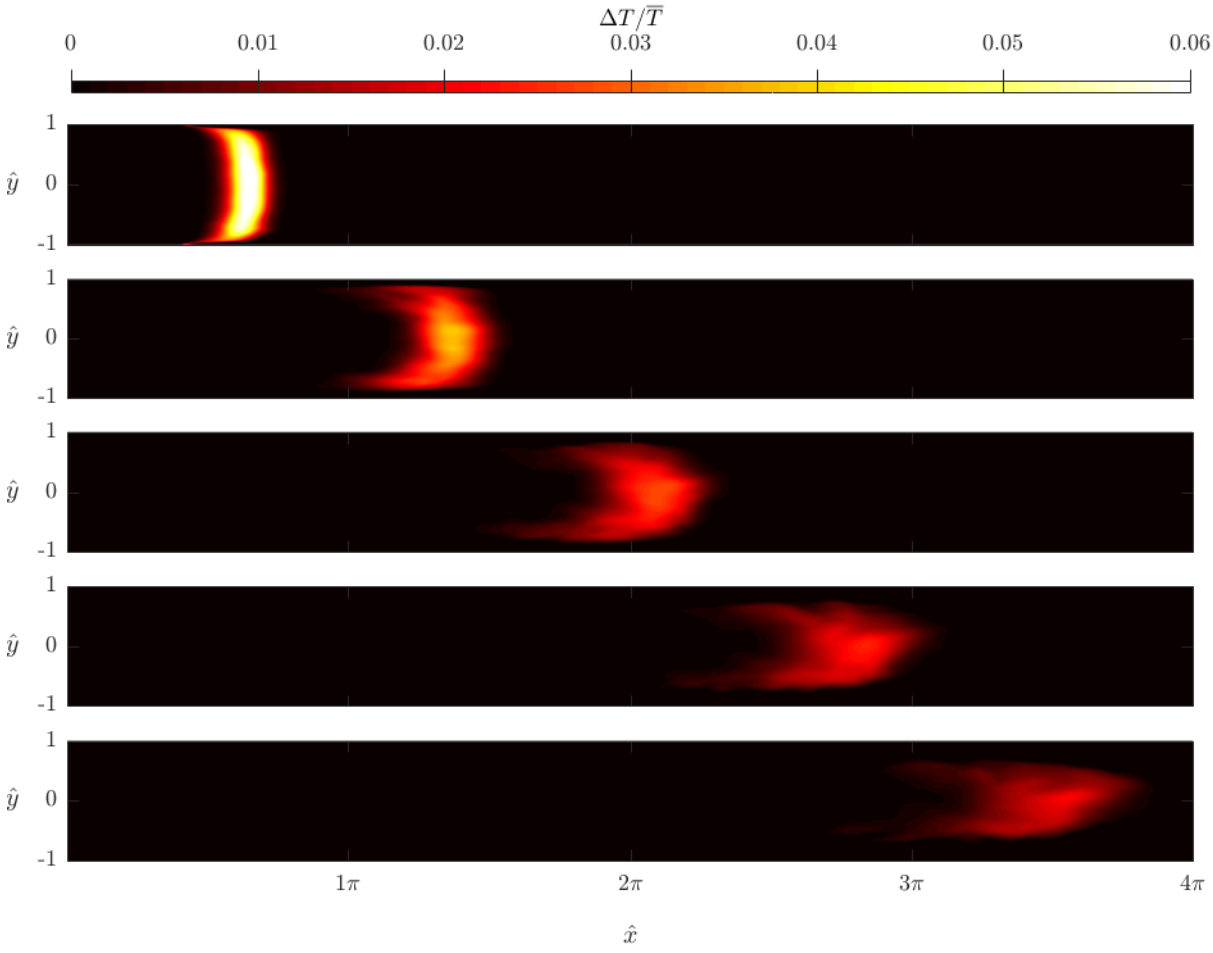

Figure 4.5: Convection of temperature perturbation through a fully developed, turbulent, channel flow with convective cooling at the walls obtained by direct numerical simulation of the high order system.

is in part a consequence of the former. Heat loss at the walls causes amplitude decay directly by removing heat from the hot fluid to the external cooling flow but also indirectly through its influence upon the shape and thickness of the wave.

The uniform shape of the wave that is imposed at the insert plane is lost due to the spatially variable velocity field. The slower mean velocity near the walls causes a notable spreading of the wave in the near wall regions and hot and cold regions of the flow co-penetrate each other giving rise to temperature gradients in the wall-normal direction that enhance the molecular diffusion of heat and thus, amplitude decay. The destructive effect of the variable velocity field just described, which is especially notable near the walls where large velocity gradients exist is widely known as the shear dispersion mechanism in the context of entropy wave attenuation. In figure 4.5, the loss of heat at the walls causes steeper temperature gradients in the wall-normal direction. Hence, amplitude decay especially in the near wall regions is much more profound when the walls are being cooled. In fact, in this scenario the amplitude of the wave quickly drops to zero near the walls and the entropy wave begins to vanish from the walls in towards the

centreline.

The spreading or thickening of the wave as it convects downstream is due to molecular and turbulent diffusion of heat from the hot fluid to the surrounding base flow. As heat gets distributed over an increasingly larger volume the amplitude of the wave decays. Further, as the wave spreads out the heat gets distributed over ever larger wavelengths. Thus, the contribution of turbulent mixing to the overall diffusion process is progressively hindered as the wave thickens and molecular diffusion becomes the dominant diffusion mechanism. Molecular diffusion of heat is much slower than turbulent diffusion, hence as molecular diffusion becomes the dominant diffusion mechanism the rate of decay of the wave amplitude slows down. In the case of periodic perturbations, temperature gradients are reduced because of the presence of hot fluid upstream and downstream and thus, in such a scenario turbulent diffusion may be more effective.

In the present study, similarly to Fattahi et al. [52], the term dissipation will be used to refer to the decay of the wave amplitude. It is necessary to make this clarification because the definition of dissipation in the context of entropy wave attenuation is not consistent in the literature. Morgans et al. [132] defined dissipation in terms of total thermal energy, which, in an adiabatic system is conserved. Dissipation defined as such does not include the decay of amplitude due to the diffusion of heat from the hot fluid to the surrounding base flow. Thus, dissipation in terms of amplitude decay is a more generic definition because it includes amplitude decay due to thermal energy loss by sinks in the flow which keeps with the definition given by [132] but also includes amplitude decay resulting from spreading of the wave, that is, the distribution of total thermal energy over a larger volume by diffusion mechanisms.

Simulation results of an entropy wave advecting in laminar flow are included in appendix C.1 and are discussed and compared with the results of the turbulent flow case.

4.3 Low order modelling

The result of the direct numerical simulation show that as the entropy wave advects through the turbulent channel flow it does not retain the uniform shape and amplitude imposed at the insert plane. The established modelling approach has kept with the one-dimensional outlook of studies of acoustic waves [151, 132, 58]. That is, the entropy wave has been integrated over a cross-section at every time step to produce a one-dimensional entropy wave. In their approach, it is assumed that the position of the entropy wave changes at a rate equal to the bulk velocity of the isothermal flow and the modelling process concentrates exclusively on the temporal decay of the volumetric amplitude.

The current work improves on the aforementioned modelling approach in two respects. Firstly, it avoids making any assumption regarding the wave speed and includes the wave position in the modelling process. Secondly, the entropy wave is not reduced to a one-dimensional wave through integration but is instead sectioned into streamwise cross-sections to allow the po-

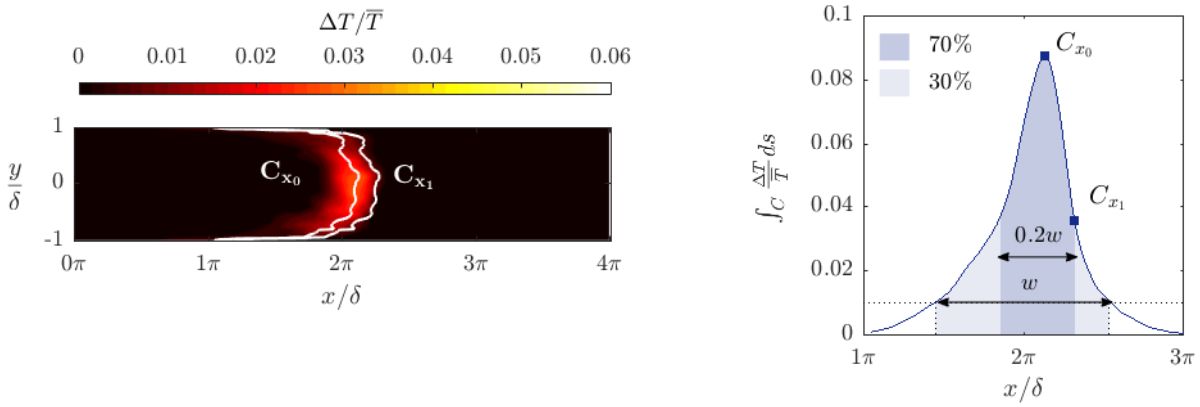

Figure 4.6: The snapshot of the entropy wave when it is at the channel half-length and the curve C connecting the positions of maximum amplitude. The amplitude is integrated along the curve as it is moved streamwise from $x/\delta = 1\pi$ to $x/\delta = 3\pi$ (measured at the centreline). The graph shows the integrated amplitude with respect to the position of the curve. A large proportion of the total thermal energy is concentrated around the positions of maximum amplitude.

sition and amplitude in each cross-section to be modelled. Hence, in the present study the terms position and amplitude refer to the positions and amplitudes from all the streamwise cross-sections, collectively.

In any streamwise cross-section of the entropy wave, the thermal energy at any moment in time is mainly concentrated around the position of the maximum amplitude in that section as shown in figure 4.6. The snapshot of figure 4.6a is from the direct numerical simulation of the adiabatic walls case and shows the state of the entropy wave once it has reached the channel half-length. The curve C_{x_0} seen in the snapshot traces the positions of maximum amplitude in the streamwise cross-sections. The integrated amplitude along C_{x_0} is plotted as a point on the graph in the right column of figure 4.6. Moving the curve C_{x_0} upstream and downstream from its original position $x/\delta = x_0$ (measured at the centreline) and integrating along it at every new position, then, plotting the integral amplitudes gives the bell-like curve seen in the graph in the right column of figure 4.6. The integral amplitude along the curve C_{x_1}, when the curve is moved to position $x/\delta = x_1$, is also plotted as a point in the graph to facilitate in making clear how the bell-curve in the graph is constructed. The bell-like distribution shows how the thermal energy is distributed around the positions of maximum amplitude (i.e. around C_{x_0}). Approximately 70% of the thermal energy is concentrated around the positions of maximum amplitude over a distance that is only 20% of the wave thickness (the thickness of the wave being defined in this case as the distance between the curves that have an integrated amplitude that is 10% of the peak amplitude $A = 0.1$). Hence, without much loss of generality the entropy wave at any moment in time can be represented simply and conveniently by the curve connecting the positions of maximum amplitude with the amplitude of the wave being the magnitude of the maximum amplitudes. Hereafter, any reference made to the position and amplitude of the entropy wave refers to the position and magnitude of the maximum amplitudes in the streamwise sections.

Time series of the position and magnitude of the maximum amplitude in each streamwise cross-section of the flow are generated from the direct numerical simulation. At every time step of the DNS the amplitudes at all nodes in the streamwise direction with the same wall-normal coordinate y are compared and the maximum amplitude in the streamwise section is found. The position at which the maximum temperature occurs is x_{max} and its magnitude is $(\Delta T/\overline{T})_{max}$. Both these quantities are a function of the wall-normal coordinate y and the instant of time at which they are evaluated. The present study used 128 nodes in the y- or wall-normal direction and therefore, generated 128 time series of x_{max} and 128 time series of $(\Delta T/\overline{T})_{max}$.

In the methodology that follows, the subscript $\{\quad\}_{max}$ is dropped from x_{max} and $(\Delta T/\overline{T})_{max}$. Furthermore, the quantity $(\Delta T/\overline{T})_{max}$ is simply denoted as T. Results are presented using the non-dimensional quantities in equation 4.1, where the caret $\{\ \hat{}\ \}$ indicates a non-dimensional quantity, τ is non-dimensional time (see equation 2.31), and δ is the channel half-height.

$$\hat{t} = \tau, \quad \hat{x} = \frac{x_{max}}{\delta}, \quad \hat{y} = \frac{y}{\delta}, \quad \hat{T} = \left.\frac{\Delta T}{\overline{T}}\right|_{max} \tag{4.1}$$

It is important to note that the methodology, although applied in the current study to time series from DNS, could equally be applied to time series obtained experimentally. However, in those cases where additional flow features related to combustor flows are present such as more involved fluid dynamics and chemical reactions, it is expected that the formulated system of equations will have a different form than that found in the current work.

4.3.1 Formulation for channel with adiabatic walls

The model is based on the assumption that the spatio-temporal evolution of the position and amplitude of the wave in a streamwise cross-section can be described by the generic non-linear system in equation 4.2, where the dot $\{\dot{}\}$ indicates the derivative with respect to time and T is the amplitude at streamwise position x.

$$\dot{T} = f(T,x) \tag{4.2a}$$

$$\dot{x} = g(T,x) \tag{4.2b}$$

Using the multivariate Taylor series expansion the equation 4.2 is expressed as the infinite summation in equation 4.3, where Φ_j is the state vector and J_{ij} is the Jacobian matrix. A repeated subscript implies summation over the range $i, j = 1, 2$.

$$\dot{\Phi}_j = J_{ij} \cdot \Phi_j + \ \textit{high order terms} \tag{4.3a}$$

$$\Phi_j = \begin{bmatrix} T \\ x \end{bmatrix} \tag{4.3b}$$

$$J_{ij} = \begin{bmatrix} \frac{\partial \dot{T}}{\partial T} & \frac{\partial \dot{T}}{\partial x} \\ \frac{\partial \dot{x}}{\partial T} & \frac{\partial \dot{x}}{\partial x} \end{bmatrix} \tag{4.3c}$$

The high order terms in equation 4.3 are neglected and the Jacobian derivatives J_{ij} are estimated from the discrete DNS data using the forward differencing scheme in equations 4.4*a-c*. Here, Δt is the simulation time step, Φ_i^t are the states at time t, and T^t and x^t are the amplitude and position of the wave respectively in a streamwise cross-section at time t.

$$\frac{\partial \Phi_j}{\partial t} \approx \frac{\Delta \Phi_j}{\Delta t} = \frac{\Phi_j^{t+\Delta t} - \Phi_j^t}{\Delta t} \tag{4.4a}$$

$$\frac{\partial \dot{\Phi}_j}{\partial T} \approx \frac{\Delta \dot{\Phi}_j}{\Delta T} = \frac{\dot{\Phi}_j^{t+\Delta t} - \dot{\Phi}_j^t}{T^{t+\Delta t} - T^t} \tag{4.4b}$$

$$\frac{\partial \dot{\Phi}_j}{\partial x} \approx \frac{\Delta \dot{\Phi}_j}{\Delta x} = \frac{\dot{\Phi}_j^{t+\Delta t} - \dot{\Phi}_j^t}{x^{t+\Delta t} - x^t} \tag{4.4c}$$

The Pearson correlation $corr(J_{ij}, \Phi_i)$ between the estimated Jacobian derivatives J_{ij} and the states Φ_j is tabulated in table 4.1. It is noted that the Jacobian derivative J_{12} (gradient of the decay rate) is strongly correlated with both amplitude and position. Correlation with both states opposed to correlation with a single state is expected because amplitude and position are not independent. That is, in the absence of heat sources the amplitude decays as the wave convects downstream and therefore the two states must be inversely proportional. Hence, J_{12} correlates positively with T but negatively with x. The fact that the Jacobian derivative J_{12} is not constant is an indication that the the dynamics of amplitude decay are non-linear. The Jacobian derivatives J_{11}, J_{21} and J_{22} are poorly correlated with the states. This suggests that they can be assumed constant, which includes the possibility of them being zero. It is important to highlight that if J_{21} and J_{22} are non-zero, then the wave speed is a function of amplitude T and position x, as these terms couple the wave speed $\dot{x}$ to the states. If J_{21} and J_{22} are zero the wave speed is constant. From the result of the DNS shown in figure 4.4, it appears that the wave speed is approximately constant and therefore there is the expectation that J_{21} and J_{22} are zero.

The wave speed is confirmed constant by the linear relation of the position with respect to time in figure 4.7*a*. The non-zero position at $\hat{t} = 0$ is the streamwise location of the insert plane at which the entropy wave is added to the flow. The time series of position used for the plot in figure 4.7*a* is from the streamwise cross-section at $\hat{y} = 0.17$. The slope of the line in figure 4.7*a* is the non-dimensional wave speed in the streamwise cross-section at $\hat{y} = 0.17$ and is $\hat{\dot{x}} \approx 15.5$. This corresponds to a velocity non-dimensionalized with respect to the friction

ij	$corr\{J_{ij},T\}$	$corr\{J_{ij},x\}$
11	-0.1	0.12
12	0.62	-0.47
21	-0.13	0.12
22	-0.08	0.03

Table 4.1: Pearson correlation coefficient of Jacobian derivatives (J_{ij}) and states (x,T). The arrow points to the Jacobian derivative that correlates well with the states.

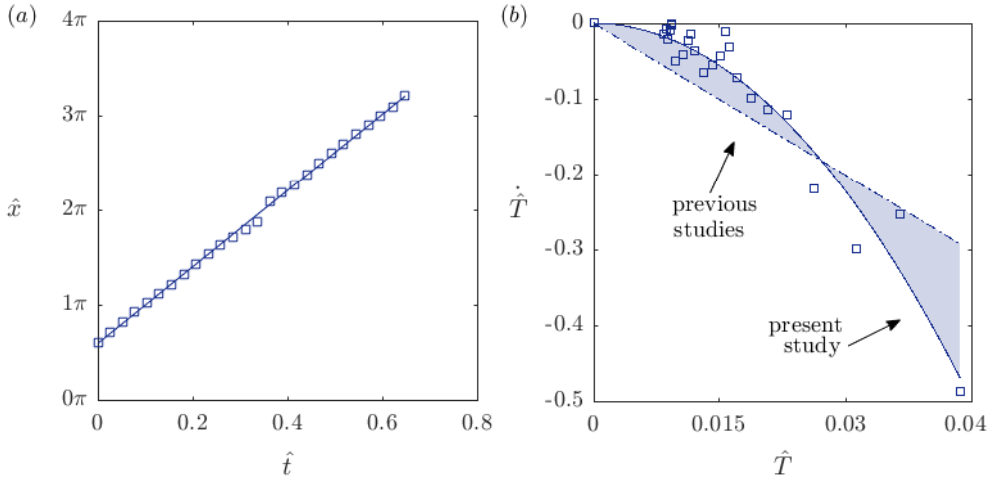

Figure 4.7: Plots of (a) the position of the maximum temperature perturbation with respect to time and (b) the decay rate with respect to amplitude in a streamwise cross-section near the centreline for the case of adiabatic walls: □ DNS, -·- linear fit, — quadratic fit

velocity $\dot{x}^+ \approx 19.5$. In figure 4.2, the mean velocity of the isothermal flow (without the entropy wave) at $\hat{y} = 0.17 \equiv y^+ \approx 150$ is $\overline{u}_1^+ = 18.3$. Hence, the wave speed $\dot{\hat{x}}$ in the streamwise cross-section at $\hat{y} = 0.17$ is within 7% of the mean velocity of the isothermal flow. This confirms the assumption made in previous studies [151, 132] that the entropy wave may be treated as a passive scalar. Nonetheless, for perturbations with peak amplitudes much larger than $A = 0.1$, the validity of the passive scalar assumption is dubious as evidence [72] exists that the entropy wave does impact on flow hydrodynamics. The data points in figure 4.7a show a linear trend. The function obtained by fitting the data with a least mean squares approach is given by equation 4.5, which is a specific form of the general equation 4.2a.

$$\dot{\hat{x}} = \theta_1(\hat{y}) \tag{4.5}$$

The hypothesis made previously that the dynamics of amplitude decay are non-linear due to observation of a non-zero correlation between the Jacobian derivative J_{12} and the states is confirmed by the plot of the time derivative of amplitude (decay rate) with respect to the amplitude shown in figure 4.7b. It should be noted that the decay rate only needs to be plotted with respect to one of the states because the states are not independent of each other and since the decay rate

is the time derivative of amplitude, it is therefore convenient to plot it with respect to amplitude. At zero amplitude the wave has completely dissipated and hence, the decay rate must also be zero. The data point at the origin is added manually to the time series taken from the DNS because the simulation is terminated once the amplitude becomes of the same order as the turbulent fluctuations at which point the wave is no longer distinguishable. The phenomenological modelling approach taken by previous studies effectively assumes that the relation between the decay rate and amplitude in figure 4.7(b) is linear. The linear fit would be the specific form of equation 4.2*b*, which, when integrated gives an amplitude decay that is exponential precisely as it is being viewed by previous studies. However, the plot in figure 4.7*b* suggests that for amplitudes $\hat{T} > 0.02$ the linear approximation underestimates the decay rate during the initial stages of the wave convection and overestimates it during the final stages. Thus, the linear approximation is capable of capturing the dynamics of the decay for amplitudes less than 2% of the base flow temperature. In contrast to previous studies, the current study captures the trend of the data points in figure 4.7*b* by a least squares quadratic fit – instead of a linear one as per equation 4.6, which provide a form for the general equation 4.2*b*. The slope of the quadratic fit in figure 4.7*b*, which is the rate of amplitude decay, decreases with the amplitude. This is keeping with the discussion of the DNS results in section 4.2. That is, turbulent and molecular transport are strong during the initial stages of the wave convection, thus causing a fast decay in the early stages of the wave advection. In the later stages, the slower molecular diffusion mechanism is dominant and results in a slower decay.

$$\dot{\hat{T}} = \theta_2(\hat{y})\hat{T}^2 \tag{4.6}$$

The above methodology produces the specific form of the general system of equations 4.2*a*-*b* given by equations 4.5 and 4.6. However, the time series came from the DNS of the channel with adiabatic walls. Applying the same methodology to the time series obtained from the DNS of the channel with heat loss at the walls reveals that the specific equations found for the adiabatic case can also be used in the case of heat loss at the walls but with a minor adjustment.

4.3.2 Adjustment for channel with cooled walls

The position of the wave in the case of heat loss at the walls is approximated well by equation 4.5 in every streamwise cross-section. However, unlike the case of adiabatic walls the amplitude decay cannot be approximated well by equation 4.6 in every streamwise cross-section. When convective heat loss is taking place at the channel walls the exponent of the term on the r.h.s. of equation 4.6 depends on the distance of the streamwise cross-section from the wall. The value of the exponent comes from the fit type that is used to approximate the relation between the $(\hat{T}, \dot{\hat{T}})$ data points. The relation transitions in this case from approximately concave-up (degree of polynomial fit is $\beta = 0.5$) in the near wall region to linear ($\beta = 1$) and then to concave-

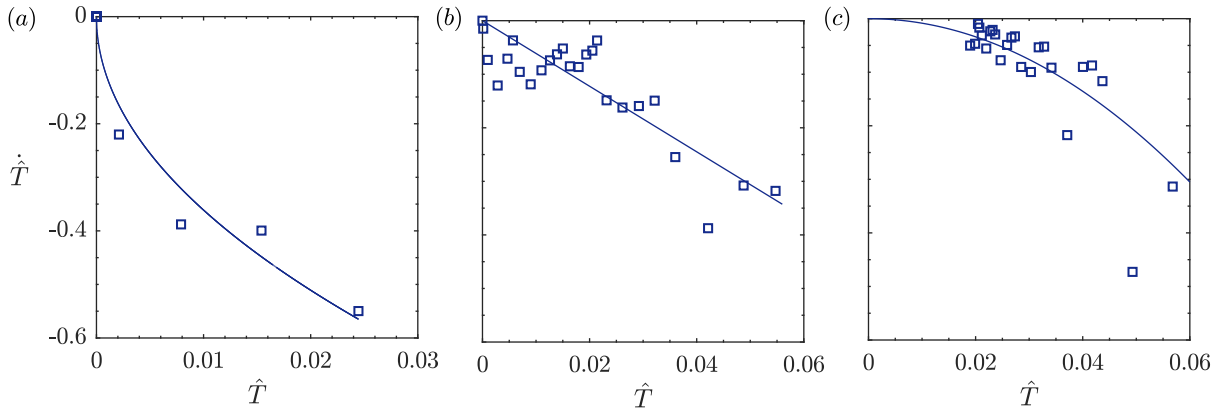

Figure 4.8: The decay rate and amplitude relation in the (a) near wall, (b) in between near wall and core flow and (c) core flow regions in the case of convective heat loss at the walls: □ DNS, —— fit

down ($\beta = 2$) in the core flow as shown in figure 4.8 for three randomly chosen streamwise cross-sections between the wall and the centreline.

The Akaike information criterion [3] or AIC provides a systematic procedure of determining the fit type (concave-up, linear, or, concave-down) that best approximates the $(\hat{T}, \dot{\hat{T}})$ data points in a particular streamwise cross-section. The AIC criterion requires that all three fit types are made to the data points and the relative likelihood that the fit is the best approximation is calculated for each fit type. The fit type with the largest relative likelihood is the best approximation in a least squares sense.

The likelihood or AIC_i value that fit type i is the best approximation is given by equation 4.7, where N is the number of data points, RSS is the residual sum of squares, and $K = 1$ in all three cases is the number of parameters in the fit type.

$$AIC_i = N\ln(RSS/N) + 2K + \frac{2K(K+1)}{N-(K+1)} \tag{4.7}$$

The relative likelihood is the likelihood with respect to the minimum as per equation 4.8, where Δ_i is the relative likelihood of fit type i.

$$\Delta_i = AIC_i - min(AIC_i) \tag{4.8}$$

It is formally customary to normalize the relative likelihood as per equation 4.9, where w_i is the normalized relative likelihood of fit type i also known as the Akaike weight.

$$w_i = \frac{\exp\left(-\frac{1}{2}\Delta_i\right)}{\sum_{i=1}^{N_{fits}} \exp\left(-\frac{1}{2}\Delta_i\right)} \tag{4.9}$$

The Akaike weight of each of the fits shown in figure 4.8 is shown in figure 4.9 for all the streamwise cross-sections. For example, very close to the walls ($\hat{y} \approx \pm 1$) the $(\hat{T}, \dot{\hat{T}})$ data points

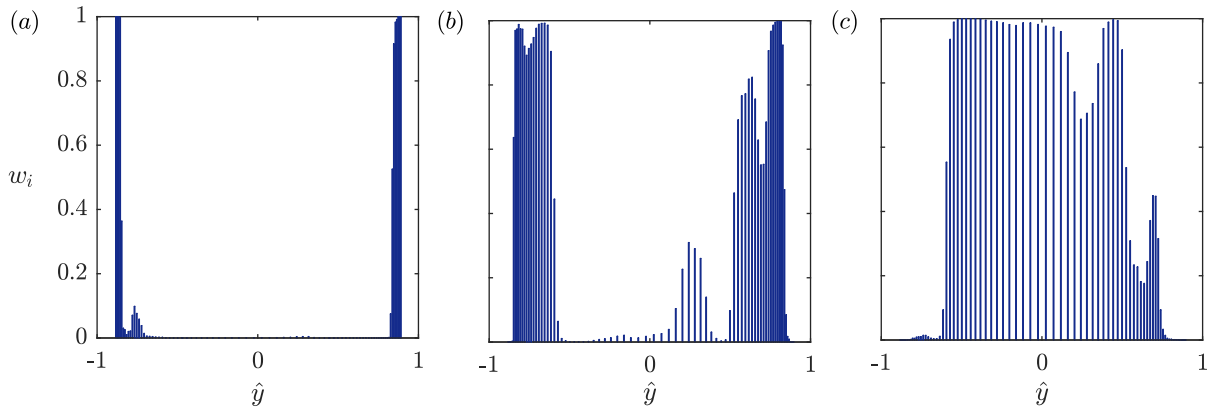

Figure 4.9: The regions of the flow in which the amplitude decays according to (*a*) $\dot{\hat{T}} = \theta_2 T^{0.5}$, (*b*) $\dot{\hat{T}} = \theta_2 T$, and (*c*) $\dot{\hat{T}} = \theta_2 T^2$

are best approximated by the fit $\dot{\hat{T}} = \theta_2 T^{0.5}$ because the Akaike weight of this fit type shown in figure 4.9*a* is very near to unity close to the walls and zero in all other streamwise cross-sections. The distribution of the Akaike weights over the channel cross-section in figure 4.9 also makes clear the thickness of the fit type 'regions'.

4.3.3 Generic formulation

The low order model (LOM) formulated for the case of the channel with adiabatic walls (equations 4.5 and 4.6) can be generalised to equation 4.10, where β depends on the thermal boundary condition at the walls.

$$\dot{\hat{x}} = \theta_1(\hat{y}) \tag{4.10a}$$

$$\dot{\hat{T}} + \theta_2(\hat{y})\hat{T}^{\beta} = 0 \tag{4.10b}$$

For adiabatic walls $\beta = 2$ in all streamwise cross-sections. In the case of convective heat loss at the walls, $\beta \in \{0.5, 1, 2\}$ depending on the distance of the streamwise cross-section from the wall. For each streamwise cross-section, a least squares fit needs to be made to the $(\hat{T}, \dot{\hat{T}})$ data points for each value of β and the best suited fit then determined using a method such as the Akaike information criterion [3].

The LOM is amenable to analytic solution. In each streamwise cross-section, the model parameters θ_1 and θ_2 are constants and the equations are decoupled and can be solved independently. The analytical solutions of equations 4.10*a* and 4.10*b* for different values of $\beta(\hat{y})$ are tabulated in table 4.2.

The model parameters $\theta_1(\hat{y})$ and $\theta_2(\hat{y})$ are not the same in all streamwise cross-sections and are therefore functions of the wall-normal coordinate $\hat{y}$. Due to the way they appear in equation 4.10, they are nominally the non-dimensional wave speed and dissipation factor, respectively. By consideration of equation 4.1 and dimensional homogeneity, the scaling factors of $\theta_1(\hat{y})$ and

β	solution of equation 4.10a	solution of equation 4.10b
$\neq 1$	$\hat{x}(\hat{t}) = \hat{x}(0) + \theta_1(\hat{y})\hat{t}$	$\hat{T}(\hat{t}) = \hat{T}(0) + \{\,[\beta(\hat{y}) - 1]\,\theta_2(\hat{y})\hat{t}\,\}^{1/[1-\beta(\hat{y})]}$
$= 1$	$\hat{x}(\hat{t}) = \hat{x}(0) + \theta_1(\hat{y})\hat{t}$	$\hat{T}(\hat{t}) = \hat{T}(0) exp\,[\,-\theta_2(\hat{y})\hat{t}\,]$

Table 4.2: Analytical solutions of the low order model given by equation 4.10.

$\theta_2(\hat{y})$ in equation 4.10 are $1/\overline{U}_{bulk}$ and $L/\overline{U}_{bulk}$, respectively. Further, since $\theta_1(\hat{y})$ and $\theta_2(\hat{y})$ are calculated through regression (of DNS data in this case), they are empirical and therefore case specific. Unlike previous models, the parameters of the current model are a function of the wall-normal coordinate and thus, the solution of the model is a position and an amplitude that are functions of the wall-normal coordinate. That is, the position and amplitude of the wave at any time are a function of the wall-normal coordinate. Hence, the model describes a two-dimensional wave. The profiles of $\theta_1(\hat{y})$ and $\theta_2(\hat{y})$ over the cross-section of the channel with adiabatic walls are shown in figure 4.10 for the turbulent Reynolds number $Re_{bulk} = 5600$ that is used in the current work and also for the laminar Reynolds numbers $Re_{bulk} = 500$ and $Re_{bulk} = 1000$.

The wave speed in a streamwise section has been found to be approximately the same (within 7%) as the mean velocity of the isothermal flow. Hence, it is expected that the cross-sectional profile of $\theta_1(\hat{y})$ will be identical to the velocity profile of the isothermal flow. This is confirmed by the plot in figure 4.10a that shows the profile of $\theta_1(\hat{y})$ over the cross-section for the turbulent flow considered in the current work and also for two laminar flows to further support the argument. For the turbulent Reynolds number, it is $\theta_1(\hat{y}) \approx 1$ over most of the cross-section except near the walls and for the laminar Reynolds numbers, the maximum is $\theta_1(0) = 1.5$ at the centreline and the profiles are parabolic. The aforementioned are indicators of the typical mean velocity profiles in turbulent and laminar channel flow.

The non-dimensional dissipation factor $\theta_2(\hat{y})$ in figure 4.10b is a minimum at the channel centreline and increases on approach to the walls for all Reynolds numbers. This keeps with observation of the DNS result shown in the left column (adiabatic walls case) of figure 4.4 where the wave clearly dissipates faster in the near wall regions relative to near the centreline because shear dispersion is stronger near the walls. Similarly, overall dissipation is stronger at the laminar Reynolds numbers relative to that at the turbulent Reynolds number because shear dispersion is stronger in laminar flows due to the less uniform velocity profile.

4.3.4 Calibration

The accuracy of the formulated LOM in equation 4.10 is evaluated by comparison of the solution with the DNS result. The DNS result are the time series $\hat{x}_{DNS}$ and $\hat{T}_{DNS}$ used to formulate the

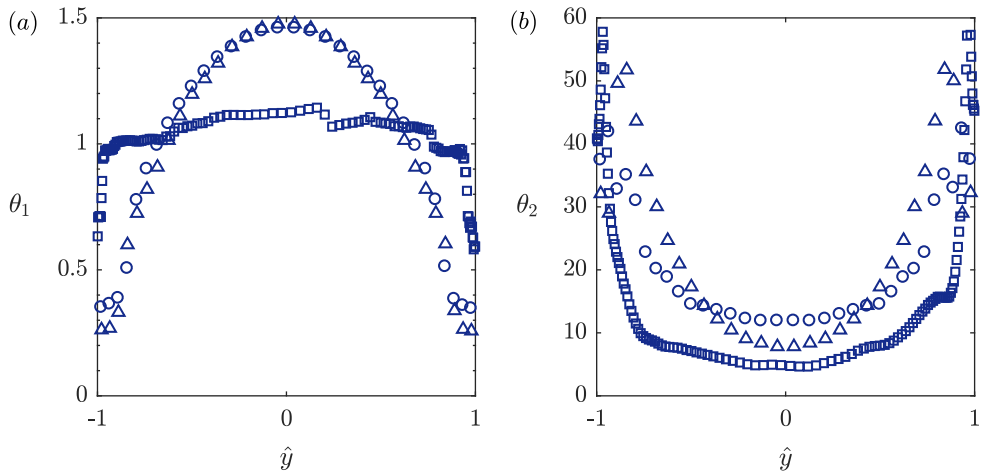

Figure 4.10: The cross-sectional profiles of the model parameters (*a*) θ_1 and (*b*) θ_2 for laminar and turbulent Reynolds numbers: $\circ$ $Re_{bulk} = 500$, $\triangle$ $Re_{bulk} = 1000$, $\square$ $Re_{bulk} = 5600$

LOM. Because the DNS result is discrete in time the time-continuous solution of the LOM (table 4.2) will be discretised to facilitate comparison of the solutions.

First, the model parameters $\theta_1(\hat{y})$ and $\theta_2(\hat{y})$ are calculated using the entire length of the $\hat{x}_{DNS}$ and $\hat{T}_{DNS}$ time series from the DNS. That is, in every streamwise cross-section, all the DNS data points are used to make the least squares fittings that give the LOM parameters $\theta_1(\hat{y})$ and $\theta_2(\hat{y})$ (as done for example in figure 4.7 for a streamwise cross-section near the centreline). Using shorter DNS time series to determine $\theta_1(\hat{y})$ and $\theta_2(\hat{y})$ will diminish the accuracy of the LOM because the fittings will in such a case be made to DNS data that does not cover the complete dynamical behaviour of the system. Thus, for the purpose of LOM calibration, the entire length of the DNS time series has to be used because in this case poor performance of the LOM will signal failure of the formulated equations and will not be due to poor estimation of the model parameters. The minimum length of the DNS time series needed for a reasonably accurate LOM will be determined in section 4.3.5. Here, the aim is to evaluate calibrate the formulated LOM and so the entire length of the DNS data is being used. Once $\theta_1(\hat{y})$ and $\theta_2(\hat{y})$ are determined they are substituted together with the initial condition $\hat{x}_{DNS}(\hat{t}=0)$ and $\hat{T}_{DNS}(\hat{t}=0)$ into the analytical solution of equation 4.10 (table 4.2) to give the time-continuous solution. The continuous solution is then sampled using a sampling period equal to the DNS time step to give time series $\hat{x}_{LOM}$ and $\hat{T}_{LOM}$ from the LOM that have a one-to-one correspondence with the time series $\hat{x}_{DNS}$ and $\hat{T}_{DNS}$ from the DNS. It should be made clear that time series $\hat{x}_{LOM}$ and $\hat{T}_{LOM}$ are obtained for each streamwise cross-section because $\theta_1(\hat{y})$ and $\theta_2(\hat{y})$ are different in each streamwise cross-section, which, is emphasized by consistently denoting their dependence on the wall-normal coordinate $\hat{y}$.

A visual comparison of the LOM and DNS solutions for the case of adiabatic walls is shown in figure 4.11. The state of the entropy wave at $\hat{t}_0$ is the initial condition $(\hat{x}_{DNS}, \hat{T}_{DNS})_{\hat{t}=0}$ and is the state immediately after the entropy wave has been added to the flow. The position of the

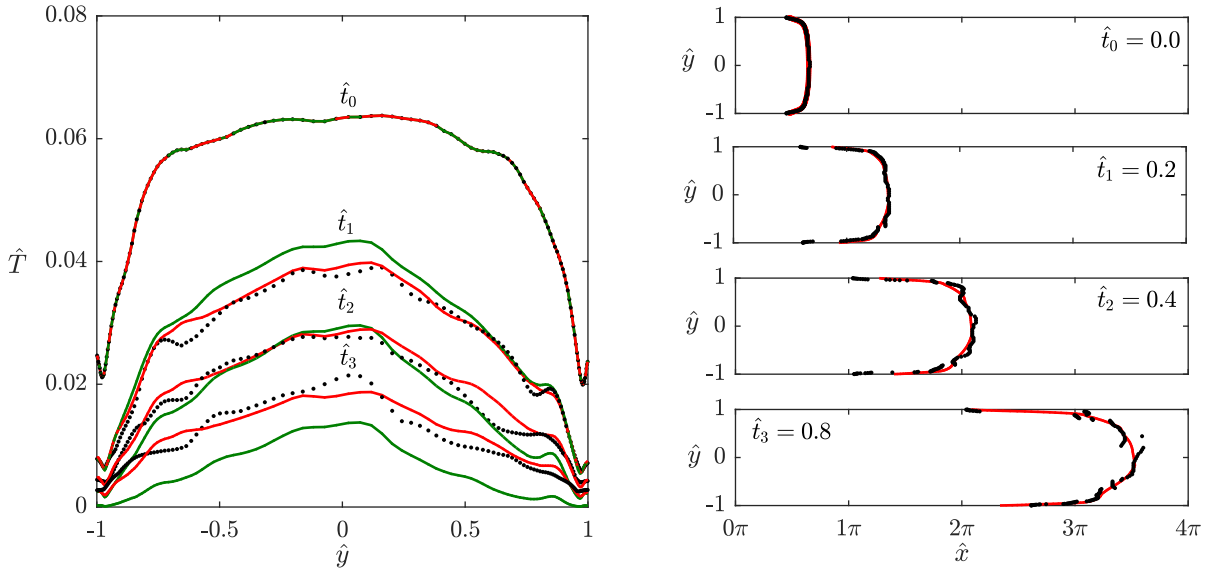

Figure 4.11: Accuracy of the low order model solution in the case of adiabatic walls: • DNS, — linear low order model, — non-linear low order model

wave in the right column of figure 4.11 is clearly in good agreement with the DNS result over the entire cross-section. In the left column of figure 4.11, the amplitude decay from both the linear ($\beta = 1$) and non-linear ($\beta = 2$) LOM is shown. The result from the linear LOM is shown due to its relevance to previous studies that assume the dynamics of amplitude decay to be linear. It is clear that the proposed non-linear LOM ($\beta = 2$) is in better agreement with the DNS result. The linear model overestimates the amplitude during the early stages of the wave convection and underestimates it during the final stages. Thus, keeping with the discussion of figure 4.7 in which the linear approximation underestimates the decay rate when the wave amplitude is large (at the early stages of the wave convection) and overestimates it when the wave amplitude is low (at the late stages of the wave convection).

A visual comparison of the LOM and DNS solutions for the case of heat loss at the walls is shown in figure 4.12. Similarly to the case of adiabatic walls, the position of the wave from the LOM is in good agreement with the position from the DNS in the right column of figure 4.12. The solution shown in figure 4.12 has a one-to-one time correspondence with the solution shown in figure 4.11 for the case of adiabatic walls. Therefore, by comparison of the wave position from the two cases it is clear that the wave fronts are convecting at the same speed in the two cases. Thus, the wave speed has not been affected by the heat loss to the walls. Hence, it is not surprising that agreement between the position of the wave from the LOM and the DNS is just as good as in the case of adiabatic walls. The heat loss to the walls has, relative to the adiabatic wall case, simply accelerated the dissipation of the wave just above the walls causing the amplitude to quickly fall to zero and the edges of the wave to vanish. In the left column of figure 4.12, the solution of the LOM equation 4.10*b* with a piecewise β (for which β is determined by the Akaike weight criterion) gives a good approximation of the amplitude decay

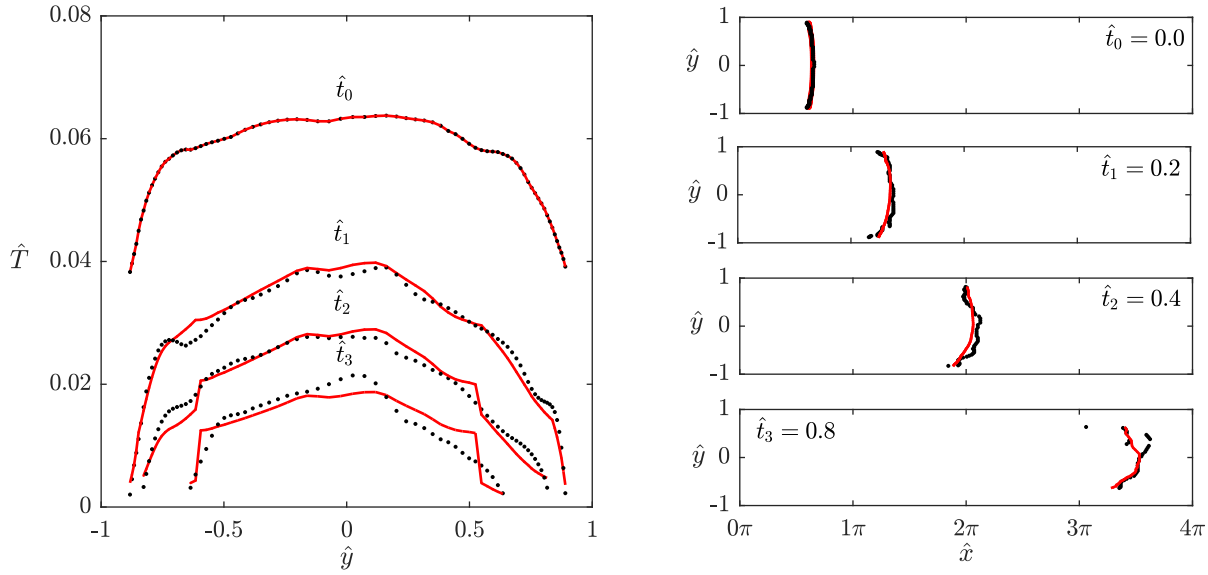

Figure 4.12: Accuracy of the low order model solution in the case of convective heat loss at the walls: • DNS, — non-linear low order model with piecewise amplitude decay

from the DNS.

The need for a piecewise β in equation 4.10*b* in the case of heat loss at the walls is emphasized in figure 4.13, which, shows the amplitude decay from the LOM when the same β is used in equation 4.10*b* for all the streamwise cross-sections. The amplitude from the equation with $\beta = 0.5$ shows better agreement with the amplitude from the DNS over the equations with $\beta = 1$ and $\beta = 2$ at the edges of the wave. On the other hand, the nonlinear equation with $\beta = 2$ shows the best agreement with the amplitude from the DNS near the centreline. Between the near-centreline and near-edge parts of the wave, the amplitude from the equation with $\beta = 1$ makes for a gentle transition. Hence, unlike the adiabatic walls case for which $\beta = 2$ could be used everywhere, in the case of heat loss at the walls it is clearly needed to use a piece-wise β over the channel cross-section.

Visual comparison of the LOM and DNS solutions through the discussion of figures 4.11 and 4.12 shows that the accuracy of the formulated LOM is good. However, a visual comparison is not made at all times because such an undertaking is tedious and impractical and as a result the deviation occurring in between the chosen times is overlooked. In order to carry out a more detailed comparison of the LOM and DNS solutions there is a need to define quantitative measures or metrics of accuracy. Such metrics are also needed in section 4.3.5 to consistently evaluate the accuracy of the LOM in relation to the length of the DNS time series. There are two criteria that have to be met for the LOM and DNS solutions to be considered in good agreement. First, the residual between the states has to be small and secondly the way in which the states vary over the cross-section has to be similar. Thus, two metrics are needed, one to measure the residual and another the correlation between the corresponding states from the LOM and the DNS.

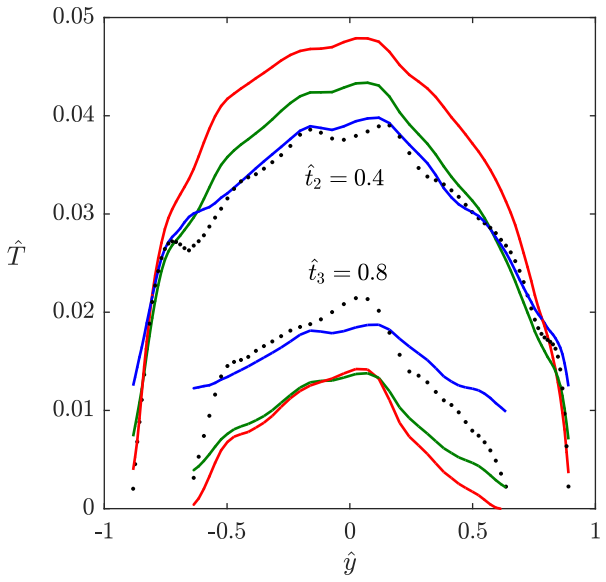

Figure 4.13: Amplitude decay from the low order model when the same β is used for all the streamwise cross-sections: • DNS, — LOM with $\beta = 0.5$, — LOM with $\beta = 1$, — LOM with $\beta = 2$

The residual between the states $\hat{\Phi}_j^{LOM}$ from the LOM and the states $\hat{\Phi}_j^{DNS}$ from the DNS is averaged over the cross-section to give an overall residual. The average residual of the states is equivalent to the residual of the average states as per equation 4.11, where $\varepsilon_{\langle\hat{\Phi}_j\rangle_A}$ is the residual of the average state $\langle\hat{\Phi}_j\rangle_A$ over the channel cross-section as per equation 4.12. The residual given by equation 4.11 is normalized with respect to the average state from the DNS. Due to the equivalence of the average residual with the residual of the average states, henceforth, the average residual over the channel cross-section is referred to simply as the residual meaning the residual of the average states. The sign of the residual indicates if the average state from the LOM is larger (positive) or smaller (negative) than the corresponding average state from the DNS.

$$\varepsilon_{\langle\hat{\Phi}_j\rangle_A} = \frac{\left\langle\hat{\Phi}_j^{LOM}\right\rangle_A - \left\langle\hat{\Phi}_j^{DNS}\right\rangle_A}{\left\langle\hat{\Phi}_j^{DNS}\right\rangle_A} \tag{4.11}$$

$$\langle\hat{\Phi}_j\rangle_A = \frac{1}{NY-1}\sum_{n=1}^{NY-1}\left(\hat{\Phi}_j\hat{y}\right)_n \tag{4.12}$$

The correlation between the states $\hat{\Phi}_j^{LOM}$ from the LOM and the states $\hat{\Phi}_j^{DNS}$ from the DNS over the channel cross-section is measured with the Pearson correlation coefficient $r_{\hat{\Phi}_j}$ as per equation 4.13. A Pearson correlation coefficient $r_{\hat{\Phi}_j} = +1$ indicates that the state from the LOM and the corresponding state from the DNS are perfectly correlated and thus, their respective profiles over the cross-section have the same trend. On the other hand a Pearson correlation coefficient $r_{\hat{\Phi}_j} = 0$ indicates that the states are uncorrelated over the channel cross-section and

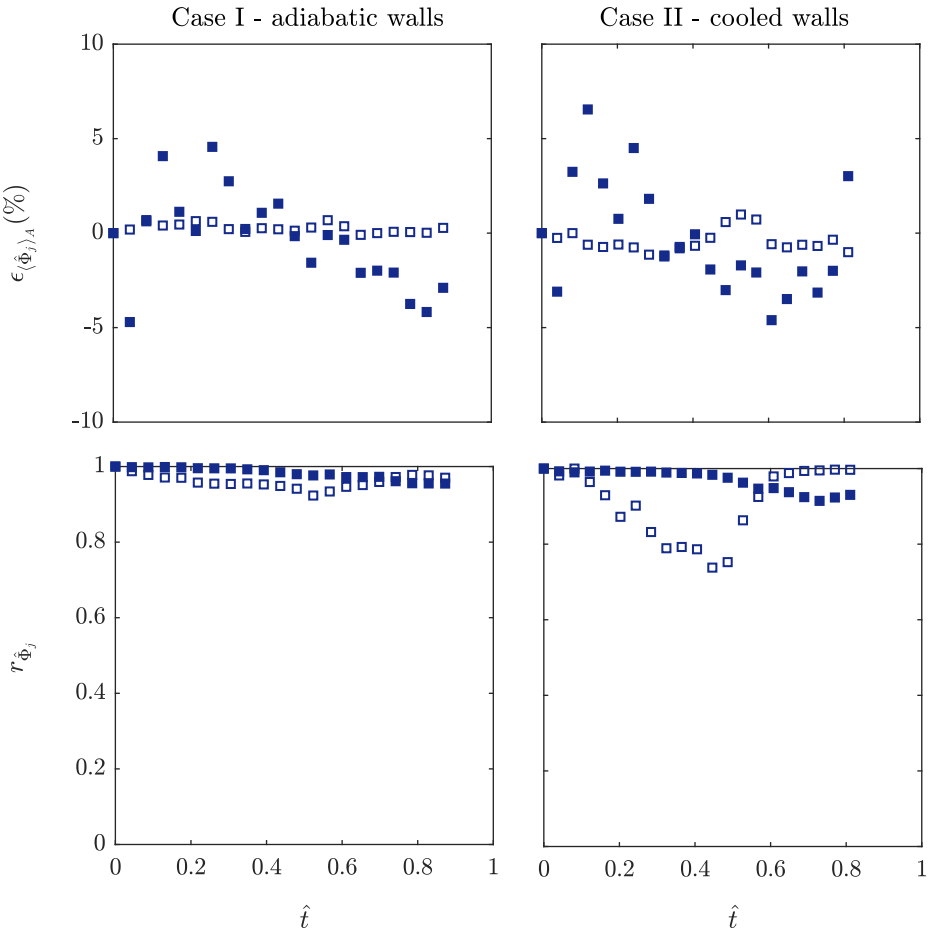

Figure 4.14: Time series plots of the residual (*row one*) and correlation (*row two*) of corresponding states from the LOM and DNS with respect to time for the case of adiabatic walls (*column one*) and heat loss at the walls (*column two*): □ $\hat{x}$, ■ $\hat{T}$

thus, that their respective profiles over the channel cross-section have radically different trends.

$$r_{\hat{\Phi}_j} = \frac{\sum_{n=1}^{NY}\left(\hat{\Phi}_j^{DNS} - \left\langle\hat{\Phi}_j^{DNS}\right\rangle_A\right)_n\left(\hat{\Phi}_j^{LOM} - \left\langle\hat{\Phi}_j^{LOM}\right\rangle_A\right)_n}{\sqrt{\sum_{n=1}^{NY}\left(\hat{\Phi}_j^{DNS} - \left\langle\hat{\Phi}_j^{DNS}\right\rangle_A\right)_n^2\sum_{j=1}^{NY}\left(\hat{\Phi}_j^{LOM} - \left\langle\hat{\Phi}_j^{LOM}\right\rangle_A\right)_n^2}} \tag{4.13}$$

Time series plots of the metrics $\varepsilon_{\langle\hat{\Phi}_j\rangle_A}$ and $r_{\hat{\Phi}_j}$ defined above are shown in columns one and two of figure 4.14 for the case of adiabatic walls and heat loss at the walls respectively.

In row one of figure 4.14, the residual is less than $\pm 5\%$ and $\pm 1\%$ at all times for amplitude and position respectively in both cases of the thermal boundary condition. The small residuals for both amplitude and position keeps with the visual comparison in figures 4.11 and 4.12. The reason for the larger amplitude residual is explained by returning to figure 4.7 in section 4.3.1, which, shows the fittings made to the DNS data in a streamwise cross-section near the channel centreline. In figure 4.7, the amplitude data from the DNS is more spread-out than the position data and this is why in row one of figure 4.14 the residual of amplitude is larger than that of

position. Furthermore, the skew-symmetry of the positive and negative residuals of amplitude in row one of figure 4.14 reflects the goodness of the fitting made to the amplitude data from the DNS during formulation of the LOM.

In row two of figure 4.14, the correlation is near unity at all times for amplitude and position in the case of adiabatic walls. In the case of heat loss at the walls, the correlation of amplitude is near unity at all times and the correlation of position, although not near unity at all times, never falls below 0.7. The correlation of the position gradually falls from close to unity at the early stages of the wave convection to near 0.7 when the wave is just past the channel half-length and then rises back to unity during the final stages of the wave convection. The snapshot at $\hat{t}_2 = 0.4$ in the right column of figure 4.12 shows the wave position from the LOM and DNS when the correlation of position is at its minimum. The position from the LOM goes from leading the position from the DNS near the upper wall to lagging it, which, creates an anti-correlation (negative correlation) over the upper part of the channel cross-section.

4.3.5 Prediction

A minimum length of the DNS time series must exist below which the LOM does not predict with acceptable accuracy the position and amplitude of the entropy wave beyond the range covered by the DNS time series used to estimate its parameters. In order to determine the minimum length of the DNS time series that is needed, the LOM parameters are estimated using progressively longer DNS time series and the accuracy of the LOM is evaluated relative to the length of the time series. The expectation is that the accuracy should approach an asymptotic limit as the length of the time series is increased from the minimum length needed for a first estimate of the model parameters to the full length of the DNS time series, which, covers the entire dynamical behaviour.

For each length of the DNS time series the accuracy of the LOM is evaluated with use of the metrics defined in section 4.3.4. However, for DNS time series of any length the metrics are time-dependent because at every new time step the states from the LOM and DNS change and thus, so does the accuracy of the LOM (see figure 4.14). Hence, the time-average of the absolute metrics ($\langle|\varepsilon_{\langle\hat{\Phi}_j\rangle_A}|\rangle_t$, $\langle|r_{\hat{\Phi}_j}|\rangle_t$) is taken as a measure of the overall accuracy of the LOM for a given length of the DNS time series. The absolute value of the metrics is taken because a positive and negative residual or correlation coefficient would cancel out when averaged and thus, the average of non-absolute metrics would give wrong indications about the LOM accuracy.

Plots of the time-averaged metrics with respect to the length of the DNS time series used to estimate the parameters of the LOM are shown in columns one and two of figure 4.15 for the case of adiabatic walls and heat loss at the walls respectively. In figure 4.15, the 0% *DNS data* corresponds to fully developed turbulent flow before the entropy wave is added to the flow. Further, the initial 12.5% of the DNS data (shaded region in the plots of figure 4.15) is the data obtained during the time taken to add the entropy wave. This initial DNS data is not used when

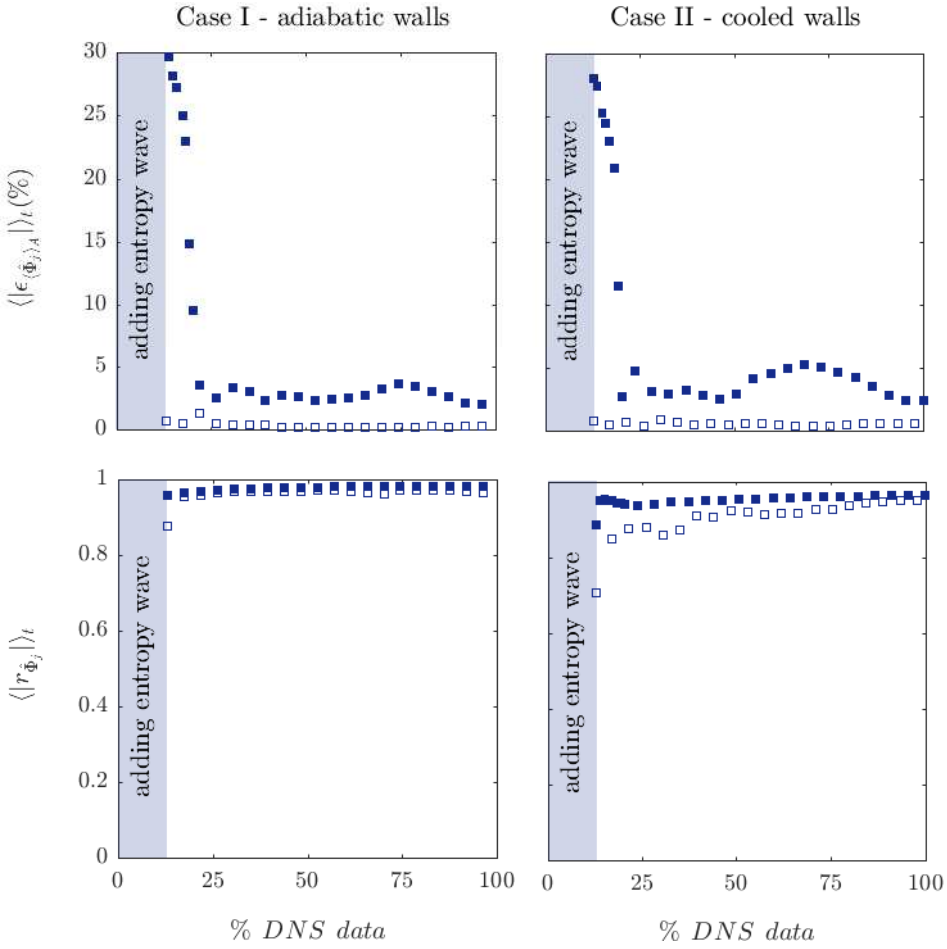

Figure 4.15: Plots of the time-averaged residual (*row one*) and time-averaged correlation (*row two*) of corresponding states from the LOM and DNS with respect to the length of the DNS time series used to formulate the LOM for the case of adiabatic walls (*column one*) and heat loss at the walls (*column two*): □ $\hat{x}$, ■ $\hat{T}$

formulating the LOM for two reasons. First, the present study is the convecting entropy wave and not its generation. Secondly, the sharp temperature gradients that exist in the flow while the entropy wave is being added to the flow would require higher order terms to be included in the LOM equations. Nevertheless, the initial 12.5% of the DNS time series is shown in the plots of figure 4.15 for completeness.

In the case of adiabatic walls in column one of figure 4.15, the residual in amplitude is large for short DNS time series because there is insufficient DNS data for a good estimate of the LOM parameters and reaches the asymptotic limit of $\approx 2\%$ at $\approx 25\%$ DNS data. As explained above, the first 12.5% of the DNS data (shaded region in the plots of figure 4.15) is not used for the estimation of the LOM parameters and thus, the effective length of the DNS time series at 25% is 12.5%. The residual in position never exceeds 2% because the wave speed is equal to the velocity of the isothermal flow (see section 4.3.1), which, is known before the entropy wave is even added to the flow. The correlation in amplitude and position is very close to unity even for short DNS time series. The correlation of position is expected to be close to unity even for

short DNS time series for the same reason as the residual is expected to be small. On the other hand, the close to unity correlation of amplitude for short DNS time series is not anticipated. It is an indication that the large residual in amplitude for short DNS time series is uniform over the channel cross-section at all times and hence, the amplitude remains correlated.

In the case of convective heat loss at the walls, in column two of figure 4.15*b*, the residual in amplitude and position are similar to the case of adiabatic walls. That is, the residual in amplitude is large for short DNS time series and reaches the asymptotic limit of $\approx 2\%$ for the model parameters estimated using only $\approx 12.5\%$ of the full length of the DNS time series. Further, the residual in position is approximately steady below 2%. The correlation in amplitude and position is very close to unity irrespective of the length of the DNS time series just as in the case of adiabatic walls.

The accuracy of the formulated LOM is acceptable only when the residual of the states is low and correlation of the states over the channel cross-section is close to unity. From the preceding discussion of figure 4.15 that an LOM of acceptable accuracy can be estimated using only 12.5% of full length of the DNS time series. Equivalently, because the full length of the DNS time series covers the complete dynamical behaviour of the convected entropy wave from insert plane to channel outlet, it can be said that an LOM of acceptable accuracy can be estimated by knowing in advance only 12.5% of the complete dynamical behaviour of the entropy wave.

4.4 Conclusions

Unlike modelling approaches of previous studies on convecting entropy perturbations, in the present study the perturbations have been given a more intuitive wave-like representation whereby the entropy wave has a non-uniform shape and amplitude. Doing so has enabled the low order modelling of the convecting entropy wave by considering it as a stand-alone dynamical system that has two states, namely position and amplitude. The rest of the flow around the wave simply being labelled as the surroundings.

For the modelling of the wave, time series of the position and amplitude of a convecting entropy wave in fully developed turbulent channel flow were generated from DNS. These time series were used to formulate and subsequently calibrate the proposed LOM through a rigorous and novel methodology that involved making statistical fittings to the DNS data. In the process, findings emerged through the methodology in support of and in opposition to the main assumptions made by the established modelling approaches of convecting small amplitude entropy waves. The first of these assumptions is that the wave may be treated as a passive scalar and the second is that the dynamics governing the decay of the wave amplitude are linear. Both assumptions are made routinely and are for the case of a wave with a small amplitude without any proof of what constitutes a small amplitude.

In support of the first assumption, the representation of the wave in the current work has

made it possible to quantitatively confirm, from the DNS data, that an entropy wave with an initial amplitude that is 10% of the base flow temperature convects at the velocity of the isothermal flow. Hence, the entropy wave may be treated as a passive scalar. However, it is emphasized that this is not to suggest that this is also the case for waves with amplitude larger than the one used in the current work.

Regarding the second assumption, the current work has found opposing evidence. The differential equation formulated for amplitude in the proposed LOM is non-linear. The current work also used a linear equation for the amplitude and found that the linear approximation is only appropriate for waves with an amplitude that is less than 2% of the base flow temperature.

The latter of the above two findings, that is, that the envelope of the linearity assumption may be smaller than previously assumed, could be the reason for the dissensus in the research community regarding the importance of the role of entropy waves with respect to the thermoacoustic stability of combustors. The proposed LOM demonstrates that a non-linear differential equation predicts the amplitude of the wave with small error. Moreover, that a linear differential equation for amplitude underestimates the amplitude of the wave during the final stages of the convection when the wave is near the combustor exit nozzle.

Conclusively, the proposed model similarly to the existing low order models of convecting entropy waves gives a wave that convects at the velocity of the isothermal flow. However, it improves on the morphology of the given wave, which is non-uniform in both position and amplitude. Furthermore, it improves on the characterisation of the amplitude decay by using a non-linear equation. The case-specific parameters of the proposed model can be estimated from limited numerical or experimental data. In the current work, the model parameters are determined using numerical data that covers only 12.5% of the complete high order dynamical behaviour. That is, only data from the near flame region is needed to calibrate the model. This, together with the fact that the model equations are amenable to analytic solution makes the proposed model suitable for integration into active control systems as a constituent part of thermoacoustic network models.

The addition of more fluid dynamics and chemical reactions could result in a different form of the model equations. For a given flow configuration (geometry, fluid dynamics, chemical reactions), the model methodology should be applied using data from a computational fluid dynamics simulation or experimental measurements to formulate the model equations. The simulation does not have to be one of high-fidelity (direct numerical simulation or large eddy simulation). However, the low order model will only be as accurate as the computational result from which it is derived. Once, the equations are established then these are expected to hold at different operating points and only a limited amount of numerical or experimental data from the high order system is needed at every operating point in order to determine the case-specific parameters of the model equations. The model is then capable of predicting the subsequent evolution of a two-dimensional entropy wave. The practicality of the model depends on whether

it can be calibrated using only data from the near flame region, which is usually modelled by large eddy simulations because flame models are too simplistic in most cases. The shape of the generated entropy perturbation in the current work is of Gaussian form. A generated entropy perturbation with a different shape would only change the initial condition that is used for the solution of the model equations.

Chapter 5

Conclusions

5.1 Contributions of the current work

The current work has studied the dynamics of processes that can influence the stability of lean premixed combustion systems and thus, limit the operational range of these systems that are an important technology for reducing NO_x emissions. The processes are flame flashback and the advection of entropy perturbations through the combustion chamber.

The dynamics prior to flashback in stable and unstable combustion have been investigated for the first time using nonlinear time series analysis. The trajectory of the system dynamics in phase space is reconstructed from existing pressure time series recorded in flashback experiments by the method of time delay embedding.

The recurrence patterns of the trajectory in the case of flashback in unstable combustion and the translation error of the trajectory in the case of flashback in stable combustion show that in both cases the determinism of the system dynamics increases on approaching the onset of flashback. From an academic point of view, the findings show that relevant dynamical changes occur in the flame on approaching the onset of flashback. Future investigations of flashback mechanisms could benefit from this information. Furthermore, from a practical point of view, the findings could be used for the design of early warning and active control systems that would ensure flashback does not occur in real combustors.

A direct numerical simulation has been carried out for an advecting entropy perturbation in a compressible turbulent channel flow. Unlike the direct numerical simulations in previous studies, the case of heat transferring walls is simulated in addition to the usual case of adiabatic walls. The simulation shows that heat transfer at the walls can significantly influence the decay of an entropy perturbation.

Time series of the position and amplitude of the advecting entropy perturbation generated from the direct numerical simulation have been used to develop a novel low order model. Unlike previous models, the perturbation has not been reduced to a plane wave with a position and an amplitude that are functions of time. Instead, the perturbation is reduced to a wave with a

position and an amplitude that are functions of time and the wall-normal coordinate, thus preserving the shape and amplitude distribution of the perturbation over the channel cross-section. The model has been developed on the assumption that the evolution of the position and amplitude of the wave can be represented by a general system of ordinary differential equations. In the framework of differential equations the position and amplitude of the wave are the states of the dynamical system. In the general system of equations, the time derivatives of the states are unknown functions of the states and the coefficients are unknown functions of the wall-normal coordinate. The former is represented by a first order Taylor series and the Jacobian matrix is approximated by applying fits to the data from the direct numerical simulation. The fit types determine the function between the states and their time derivatives. The fitting coefficients are the coefficients in the resulting equations and depend on the wall-normal coordinate. In a streamwise section, the equations for position and amplitude are uncoupled and hence, the time evolution of the position and the amplitude in a streamwise section can be discussed separately.

The equation for the position of the wave shows that the wave changes position at a rate equal to the velocity of the isothermal flow. This finding is for perturbations with a peak amplitude that is less than 10% of the base flow temperature, which is the peak amplitude considered in the current work.

The equation for the amplitude of the wave is nonlinear because the Jacobian depends on the states. The Jacobian is constant for a peak amplitude less than 2% of the base flow temperature. Hence, the assumption of previous studies that the governing equation for the amplitude decay is linear is only strictly valid for peak amplitudes less than 2% of the base flow temperature. For perturbations greater than this threshold, the linear equation overestimates and underestimates the amplitude during the initial and final stages of the advection process respectively.

The low order model that is developed initially for the adiabatic walls case has been extended to include the effects of heat transfer at the walls that have not been included in previous models. In this case, the fit type also depends on the wall-normal coordinate and a library of fit types is necessary so that the best fit type can be selected for a particular streamwise section based on a best fit type selection criterion.

5.2 Suggestions for further work

A future study on flame flashback could investigate the feasibility of an active control system that would detect and suppress the phenomenon. The control system would monitor the dynamics of a combustion system by reconstructing its trajectory in phase space from the time series of a measured variable such as pressure. On detecting the flashback precursors that have been identified in the current work, the control system would take measures to prevent the occurrence of flame flashback (e.g. decrease the swirl number by adjusting the swirl vanes angle). The response time of the control system could benefit from short term prediction of the system

dynamics that can be achieved by extrapolating the known trajectory prior to searching for the flashback precursors. Furthermore, the response time could benefit from optimisation of the coded algorithms that carry out the time series analysis. In the current work, code optimisation has not been a prime concern. However, efficient coding of the algorithms that carry out the time series analyses is necessary when these are intended for use in control systems.

Further work related to the modelling of advecting entropy waves could first, attempt to model the advection of larger amplitude entropy waves. The amplitude considered in the current work is 10% of the base flow temperature. The mean flame temperature in lean premixed combustors can be as low as $1600K$. The adiabatic flame temperature of most hydrocarbon flames is in excess of $2000K$. It is therefore conceivable that entropy waves with an amplitude that is as high as 25% of the base flow temperature could be generated at the flame. Recent studies have shown that entropy waves do influence the flow hydrodynamics. Although the exerted influence is subtle in the case of low amplitude entropy waves, for stronger entropy waves this may not be the case. A high amplitude entropy wave could result in the time scales of buoyancy driven processes being comparable to the mean residence time of the flow. Further, later work could also include additional physics in the flow and evaluate the model performance. The inclusion of additional physics is expected to require that the library of fit types used during the modelling methodology be expanded. Additional physics could be introduced by adding a swirl velocity component to the flow and chemical reactions. Furthermore, the turbulent intensity of the flow could be increased. The ability of the model to capture additional physics could further be extended by including the Hessian matrix and higher order terms into the formulation.

Bibliography

[1] A review of active control approaches in stabilizing combustion systems in aerospace industry. *Prog. Aerosp. Sci.*, 97:35 – 60, 2018.

[2] H. Abarbanel. The analysis of observed chaotic data in physical systems. *Rev. Mod. Phys.*, 65:1331–1392, 1993.

[3] H. Akaike. *Akaike's Information Criterion*, page 25. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[4] A. S. Alzwayi, M. C. Paul, and S. Navarro-Martinez. Large eddy simulation of transition of free convection flow over an inclined upward facing heated plate. *Int. Commun. Heat Mass Transfer*, 57:330–340, 2014.

[5] W. T. Ashurst. Flame propagation along a vortex: the baroclinic push. *Combust. Sci. Technol.*, 112(1):175–185, 1996.

[6] B. Babaei, R. Zarghami, H. Sedighikamal, R. Sotudeh-Gharebagh, and N. Mostoufi. Selection of minimal length of line in recurrence quantification analysis. *Physica A*, 395: 112–120, 2014.

[7] F. Bake, U. Michel, and I. Roehle. Investigation of entropy noise in aero-engine combustors. *ASME. J. Eng. Gas Turbines Power*, 129(2):370–376, 2006.

[8] F. Bake, N. Kings, and I. Röhle. Fundamental mechanism of entropy noise in aero-engines : Experimental investigation. *J. Eng. Gas Turbines Power*, 130(1):011202, 2008.

[9] F. Bake, C. Richter, B. Mühlbauer, N. Kings, I. Röhle, F. Thiele, and B. Noll. The entropy wave generator (ewg): A reference case on entropy noise. *J. Sound Vib.*, 326(3):574–598, 2009.

[10] R. Balachandran, B. Ayoola, C. Kaminski, A. Dowling, and E. Mastorakos. Experimental investigation of the nonlinear response of turbulent premixed flames to imposed inlet velocity oscillations. *Combust. Flame*, 143:37–55, 2005.

[11] G. Baumgartner and T. Sattelmayer. Experimental investigation of the flashback limits and flame propagation mechanisms for premixed hydrogen-air flames in non-swirling and swirling flow. volume Volume 1A: Combustion, Fuels and Emissions, pages GT2013–94258, 2013.

[12] A. C. Benim and K. J. Syed. *Flashback Mechanisms in Lean Premixed Gas Turbine Combustion*. Academic Press, 2014.

[13] A. Berlad and A. Potter. Relation of boundary velocity gradient for flash-back to burning velocity and quenching distance. *Combust. Flame*, 1(1):127–128, 1957.

[14] G. Blesinger, R. Koch, and H.-J. Bauer. Influence of flow field scaling on flashback of swirl flames. *Exp. Therm. Fluid Sci.*, 34(3):290 – 298, 2010.

[15] M. S. Bohn. *Noise produced by the interaction of acoustic waves and entropy waves with high-speed nozzle flows*. PhD thesis, California Institute of Technology, 1976.

[16] J. Brand, S. Sampath, F. Shum, R. L. Bayt, and J. Cohen. Potential use of hydrogen in air propulsion. In *AIAA International Air and Space Symposium and Exposition: The Next 100 Years*, Dayton, Ohio, USA, 2003.

[17] M. J. Brear, D. C. Carolan, and N. Karimi. Dynamic response of the exit nozzle of a premixed combustor to pressure and entropic disturbances. In *Proceedings of the 8th Euromech Fluid Mechanics Conference*, Bad Reichenhall, Germany, 2010.

[18] S. Candel. Combustion dynamics and control: progress and challenges. *Proc. Combust. Inst.*, 29(1):1–28, 2002.

[19] S. Candel, D. Durox, S. Ducruix, A.-L. Birbaud, N. Noiray, and T. Schuller. Flame dynamics and combustion noise: Progress and challenges. *Int. J. Aeroacoust.*, 8(1):1–56, 2009.

[20] S. Candel, D. Durox, T. Schuller, N. Darabiha, L. Hakim, and T. Schmitt. Advances in combustion and propulsion applications. *Eur. J. Mech. B. Fluids*, 40:87–106, 2013.

[21] S. Candel, D. Durox, T. Schuller, J.-F. Bourgouin, and J. P. Moeck. Dynamics of swirling flames. *Annu. Rev. Fluid Mech.*, 46(1):147–173, 2014.

[22] D. Carolan. Measurement of the transfer function of a combustor exit nozzle. Master's thesis, University of Melbourne, 2009.

[23] M. Chaos and F. L. Dryer. Syngas combustion kinetics and applications. *Combust. Sci. Technol.*, 180(6):1053–1096, 2008.

[24] J. Chomiak. Dissipation fluctuations and the structure and propagation of turbulent flames in premixed gases at high reynolds numbers. *Symp. (Int.) Combust.*, 16(1):1665–1673, 1977.

[25] S. M. Correa. A review of nox formation under gas-turbine combustion conditions. *Combust. Sci. Technol.*, 87:329–362, 1992.

[26] F. E. C. Culick. Unsteady motions in combustion chambers for propulsion systems. *AGARDograph*, pages NATO/RTO–AG–AVT–039, 2006.

[27] N. Cumpsty. Jet engine combustion noise: Pressure, entropy and vorticity perturbations produced by unsteady combustion or heat addition. *J. Sound Vib.*, 66(4):527 – 544, 1979.

[28] N. Cumpsty and F. Marble. Core noise from gas turbine exhausts. *J. Sound Vib.*, 54(2): 297 – 309, 1977.

[29] B. Dam, G. Corona, M. Hayder, and A. Choudhuri. Effects of syngas composition on combustion induced vortex breakdown (civb) flashback in a swirl stabilized combustor. *Fuel*, 90(11):3274 – 3284, 2011.

[30] H. Daneshyar and P. G. Hill. The structure of small-scale turbulence and its effect on combustion in spark ignition engines. 13:47–73, 1987.

[31] M. Darecki, C. Edelstenne, T. Enders, E. Fernandez, P. Hartman, J.-P. Herteman, M. Kerkloh, I. King, P. Ky, M. Mathieu, G. Orsi, G. Schotman, C. Smith, and J.-D. Wörner. Acare flightpath 2050 - europe's vision for aviation. Technical report, Advisory Council for Aeronautics Research in Europe, 2011.

[32] K. Darrow, R. Tidball, J. Wang, and A. Hampson. Catalog of chp technologies. Technical report, U.S. Environmental Protection Agency, 2015.

[33] A. De and S. Acharya. Parametric study of upstream flame propagation in hydrogen-enriched premixed combustion: Effects of swirl, geometry and premixedness. *Int. J. Hydrogen Energy*, 37:14649–14668, 10 2012.

[34] A. De and S. Acharya. Dynamics of upstream flame propagation in a hydrogen-enriched premixed flame. *Int. J. Hydrogen Energy*, 37:17294–17309, 11 2012.

[35] K. Döbbeling, J. Hellat, and H. Koch. 25 years of bbc/abb/alstom lean premixed combustion technologies. *J. Eng. Gas Turbines Power*, 129:2–12, 2007.

[36] F. D. Domenico, P. Shah, S. M. Lowe, L. Fan, P. Ewart, B. A. O. Williams, and S. Hochgreb. High frequency measurement of temperature and composition spots with litgs. *ASME. J. Eng. Gas Turbines Power*, 141(3):031003–031003–11, 2018.

[37] A. Dowling. The calculation of thermoacoustic oscillations. *J. Sound Vib.*, 180(4):557 – 581, 1995.

[38] A. P. Dowling. Acoustics of unstable flows. In T. Tatsumi, E. Watanabe, and T. Kambe, editors, *Theoretical and Applied Mechanics: Proceedings of the International Congress of Theoretical and Applied Mechanics*. Elsevier, 1996.

[39] A. P. Dowling and Y. Mahmoudi. Combustion noise. *Proc. Comb. Inst.*, 35:65–100, 2015.

[40] A. P. Dowling and S. R. Stow. Acoustic analysis of gas turbine combustors. *J. Propul. Power*, 19:751–764, 2003.

[41] I. Duran and S. Moreau. Solution of the quasi-one-dimensional linearized euler equations using flow invariants and the magnus expansion. *J. Fluid Mech.*, 723:190–231, 2013.

[42] I. Duran, M. Leyko, S. Moreau, F. Nicoud, and T. Poinsot. Computing combustion noise by combining large eddy simulations with analytical models for the propagation of waves through turbine blades. *C.R. Mec.*, 341(1):131 – 140, 2013.

[43] I. Duran, S. Moreau, and T. Poinsot. Analytical and numerical study of combustion noise through a subsonic nozzle. *AIAA J.*, 51(1):42–52, 2013.

[44] J.-P. Eckmann, S. Kamphorst, and D. Ruelle. Recurrence plots of dynamical systems. *Europhys. Lett.*, 5:973–977, 1987.

[45] J. Eckstein and T. Sattelmayer. Low-order modelling of low frequency combustion instabilities in aeroengines. *J. Propul. Power*, 22(2):425–432, 2006.

[46] J. Eckstein, E. Freitag, and C. H. T. Sattelmayer. Experimental study on the role of entropy waves in low-frequency oscillations in a rql combustor. *J. Eng. Gas Turbines Power*, 128(2):264–270, 2006.

[47] Eichler and T. Sattelmayer. Experiments on flame flashback in a quasi-2D turbulent wall boundary layer for premixed methane-hydrogen-air mixtures. *J. Eng. Gas Turb. Power*, 133(1), 2011.

[48] C. Eichler and T. Sattelmayer. Premixed flame flashback in wall boundary layers studied by long-distance micro-PIV. *Exp. Fluids*, 52(2):347–360, 2012.

[49] C. Eichler, G. Baumgartner, and T. Sattelmayer. Experimental investigation of turbulent boundary layer flashback limits for premixed hydrogen-air flames confined in ducts. In *Volume 2: Combustion, Fuels and Emissions, Parts A and B*. ASME International, 2012.

[50] B. Emerson, L. Angello, T. Lieuwen, and D. Noble. Advanced gas turbine combustor health monitoring using combustion dynamics data. 2016.

[51] M. Escudier. Vortex breakdown: Observations and explanations. *Prog. Aerosp. Sci.*, 25 (2):189–229, 1988.

[52] A. Fattahi, S. Hosseinalipour, and N. Karimi. On the dissipation and dispersion of entropy waves in heat transferring channel flows. *Phys. Fluids*, 29:087104:1–16, 2017.

[53] A. M. Fraser and H. L. Swinney. Independent coordinates for strange attractors from mutual information. *Phys. Rev. A*, 33:1134–1140, 1986.

[54] C. Frei, R. Whitney, H.-W. Schiffer, K. Rose, D. A. Rieser, A. Al-Qahtani, and P. Thomas. The world energy scenarios: Composing energy futures to 2050. Technical report, World Energy Council, 2013.

[55] J. Fritz, M. Kröner, and T. Sattelmayer. Flashback in a swirl burner with cylindrical premixing zone. 2:2001–GT–0054, 2001.

[56] J. Fritz, M. Kröner, and T. Sattelmayer. Flashback in a swirl burner with cylindrical premixing zone. *J. Eng. Gas Turbines Power*, 126(2):276–283, 2004.

[57] R. G. Gallager. *Information theory and reliable communication*. Wiley, 1968.

[58] A. Giusti, N. A. Worth, E. Mastorakos, and A. P. Dowling. Experimental and numerical investigation into the propagation of entropy waves. *AIAA J.*, 55:446–458, 2016.

[59] C. S. Goh and A. S. Morgans. Phase prediction of the response of chocked nozzles to entropy and acoustic disturbances. *J. Sound Vib.*, 330:5184–5198, 2011.

[60] C. S. Goh and A. S. Morgans. The influence of entropy waves on the thermoacoustic stability of a model combustor. *Combust. Sci. Technol.*, 185(2):249–268, 2013.

[61] C. S. Goh and A. S. Morgans. The influence of entropy waves on the thermoacoustic stability of a model combustor. *Combust. Sci. Technol.*, 185:249–268, 2013.

[62] H. Gotoda, H. Nikimoto, T. Miyano, and S. Tachibana. Dynamic properties of combustion instability in a lean premixed gas-turbine combustor. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 21:013124, 2011.

[63] H. Gotoda, Y. Shinoda, M. Kobayashi, Y. Okuno, and S. Tachibana. Detection and control of combustion instability based on the concept of dynamical system theory. *Phys. Rev. E*, 89:022910, 2014.

[64] G. A. Gottwald and I. Melbourne. Testing for chaos in deterministic systems with noise. *Physica D*, 212(1):100–110, 2005.

[65] A. Gruber, R. Sankaran, E. R. Hawkes, and J. H. Chen. Turbulent flame-wall interaction: a direct numerical simulation study. *J. Fluid Mech.*, 658:5–32, 2010.

[66] A. Gruber, J. H. Chen, D. Valiev, and C. K. Law. Direct numerical simulation of premixed flame boundary layer flashback in turbulent channel flow. *J. Fluid Mech.*, 709:516–542, 2012.

[67] A. K. Gupta, D. G. Lilley, and N. Syred. *Swirl flows*. Abacus Press, 1984.

[68] R. Hanson and A. Thomas. Flame development in swirling flows in closed vessels. *Combust. Flame*, 55(3):255–277, 1984.

[69] C. Heeger, R. L. Gordon, M. J. Tummers, T. Sattelmayer, and A. Dreizler. Experimental analysis of flashback in lean premixed swirling flames: upstream flame propagation. *Exp. Fluids*, 49(4):853–863, 2010.

[70] P. A. Hield and M. J. Brear. Comparison of open and choked premixed combustor exits during thermoacoustic limit cycle. *AIAA J.*, 46(2):517–526, 2008.

[71] P. A. Hield, M. J. Brear, and S. H. Jin. Thermoacoustic limit cycles in a premixed laboratory combustor with open and choked exits. *Combust. Flame*, 156:1683–1697, 2009.

[72] M. Hosseinalipour, A. Fattahi, A. Afshari, and N. Karimi. On the effects of convecting entropy waves on the combustor hydrodynamics. *Appl. Therm. Eng.*, 110:901–909, 2017.

[73] M. S. Howe. The generation of sound by aerodynamic sources in an inhomogeneous steady flow. *J. Fluid Mech.*, 67(3):597–610, 1975.

[74] M. Huet and A. Giauque. A nonlinear model for indirect combustion noise through a compact nozzle. *J. Fluid Mech.*, 733:268–301, 2013.

[75] M. Ihme. Combustion and engine-core noise. *Annu. Rev. Fluid Mech.*, 49(1):277–310, 2017.

[76] S. Ishizuka. Flame propagation along a vortex axis. *Prog. Energy Combust. Sci.*, 28(6): 477 – 542, 2002.

[77] S. Ishizuka, T. Hamasaki, K. Koumura, and R. Hasegawa. Measurements of flame speeds in combustible vortex rings: Validity of the back-pressure drive flame propagation mechanism. *Symp. (Int.) Combust.*, 27(1):727–734, 1998.

[78] B. N. Jean-Michel Lourier, Andreas Huber and M. Aigner. Numerical analysis of indirect combustion noise generation within a subsonic nozzle. 2014.

[79] W. Jones and M. Wille. Large-eddy simulation of a plane jet in a cross-flow. *International Journal of Heat and Fluid Flow*, 17(3):296 – 306, 1996.

[80] L. Kabiraj, A. Saurabh, P. Wahi, and R. I. Sujith. Route to chaos for combustion instability in ducted laminar premixed flames. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22:023129, 2012.

[81] L. Kabiraj, A. Saurabh, N. Karimi, A. Sailor, E. Mastorakos, and A. Dowiling. Chaos in an imperfectly premixed model combustor. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 25:023101, 2015.

[82] L. Kabiraj, A. Saurabh, H. Nawroth, C. O. Paschereit, R. I. Sujith, and N. Karimi. Recurrence plots for the analysis of combustion dynamics. In *Recurrence Plots and Their Quantifications: Expanding Horizons*, pages 321–339. Springer International Publishing, 2016.

[83] H. Kantz and T. Schreiber. *Nonlinear Time Series Analysis*. Cambridge University Press, 2004.

[84] D. Kaplan and L. Glass. Direct test for determinism in a time series. *Phys. Rev. Lett.*, 68: 427–430, 1992.

[85] N. Karimi. Response of a conical, laminar premixed flame to low amplitude acoustic forcing - a comparison between experiment and kinematic theories. *Energy*, 78:490–500, 2014.

[86] N. Karimi, M. J. Brear, and W. H. Moase. Acoustic and disturbance energy analysis of a flow with heat communication. *J. Fluid Mech.*, 597:67–89, 2008.

[87] N. Karimi, M. J. Brear, and W. H. Moase. On the interaction of sound with steady heat communicating flows. *J. Sound Vib.*, 329(22):4705 – 4718, 2010.

[88] N. Karimi, C. Heeger, L. Christodoulou, and A. Dreizler. Experimental and theoretical investigation of the flashback of a swirling, bluff-body stabilised, premixed flame. *Z. Phys. Chem.*, 229:663–689, 2015.

[89] N. Karimi, S. McGrath, P. Brown, J. Weinkauf, and A. Dreizler. Generation of adverse pressure gradient in the circumferential flashback of premixed flame. *Flow. Turbul. Combust.*, 97(2):663–687, 2016.

[90] J. J. Keller. Thermoacoustic oscillations in combustion chambers of gas turbines. *AIAA J.*, 33(12):2280–2287, 1995.

[91] J. O. Keller, L. Vaneveld, D. Korschelt, G. L. Hubbard, A. F. Ghoniem, J. W. Daily, and A. K. Oppenheim. Mechanism of instabilities in turbulent combustion leading to flashback. *AIAA J.*, 20(2):254–262, 1981.

[92] M. B. Kennel, R. Brown, and H. D. I. Abarbanel. Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Phys. Rev. A*, 45:3403–3411, 1992.

[93] D. Kershaw. The incomplete cholesky-conjugate gradient method for the iterative solution of systems. *J. Comput. Phys.*, 26:43–65, 1978.

[94] D. E. Keyes, L. C. McInnes, C. Woodward, W. Gropp, E. Myra, M. Pernice, J. Bell, J. Brown, A. Clo, J. Connors, E. Constantinescu, D. Estep, K. Evans, C. Farhat, A. Hakim, G. Hammond, G. Hansen, J. Hill, T. Isaac, X. Jiao, K. Jordan, D. Kaushik, E. Kaxiras, A. Koniges, K. Lee, A. Lott, Q. Lu, J. Magerlein, R. Maxwell, M. McCourt, M. Mehl, R. Pawlowski, A. P. Randles, D. Reynolds, B. Riviére, U. Rüde, T. Scheibe, J. Shadid, B. Sheehan, M. Shephard, A. Siegel, B. Smith, X. Tang, C. Wilson, and B. Wohlmuth. Multiphysics simulations: Challenges and opportunities. *Int. J. High Perform. Comput. Appl.*, 27(1):4–83, 2013.

[95] F. Kiesewetter, M. Konle, and T. Sattelmayer. Analysis of combustion induced vortex breakdown driven flame flashback in a premix burner with cylindrical mixing zone. *J. Eng. Gas Turbine Power*, 129:929–936, 2007.

[96] J. Kim, P. Moin, and R. Moser. Turbulence statistics in fully developed channel flow at low reynolds number. *J. Fluid Mech.*, 177:133–166, 1987.

[97] N. Kings and F. Bake. Indirect combustion noise: Noise generation by accelerated vorticity in a nozzle flow. *Int. J. Spray Combust.*, 2(3):253–266, 2010.

[98] M. Konle and T. Sattelmayer. Interaction of heat release and vortex breakdown during flame flashback driven by combustion induced vortex breakdown. *Exp. Fluids*, 47(4-5): 627–635, 2009.

[99] M. Konle and T. Sattelmayer. Time scale model for the prediction of the onset of flame flashback driven by combustion induced vortex breakdown. *J. Eng. Gas Turbines Power*, 132(4):041503, 2010.

[100] M. Konle, F. Kiesewetter, and T. Sattelmayer. Simultaneous high repetition rate PIV–LIF-measurements of CIVB driven flashback. *Exp. Fluids*, 44(4):529–538, 2008.

[101] M. Kousoulidou and L. Lonza. European aviation environmental report 2016. Technical report, 2016.

[102] M. Kröner, J. Fritz, and T. Sattelmayer. Flashback limits for combustion induced vortex breakdown in a swirl burner. *J. Eng. Gas Turbines Power*, 125:693–700, 2003.

[103] M. Kröner, T. Sattelmayer, J. Fritz, F. Kiesewetter, and C. Hirsch. Flame propagation in swirling flows - effect of local extinction on the combustion induced vortex breakdown. *Combust. Sci. Technol.*, 179(7):1385–1416, 2007.

[104] V. Kurdyumov, E. Fernández, and A. Liñán. Flame flashback and propagation of premixed flames near a wall. *Proc. Combust. Inst.*, 28(2):1883 – 1889, 2000.

[105] V. Kurdyumov, E. Fernández-Tarrazo, J.-M. Truffaut, J. Quinard, A. Wangher, and G. Searby. Experimental and numerical study of premixed flame flashback. *Proc. Combust. Inst.*, 31(1):1275–1282, 2007.

[106] K. Lam, P. Geipel, and J. Larfeldt. Hydrogen enriched combustion testing of siemens industrial SGT-400 at atmospheric conditions. *J. Eng. Gas Turbines Power*, 137(2):021502, 2014.

[107] P. Landa and M. Rosenblum. Time series analysis for system identification and diagnostics. *Physica D*, 48(1):232–254, 1991.

[108] A. H. Lefebvre and D. R. Ballal. *Gas Turbine Combustion: Alternative Fuels and Emissions*. CRC Press, 2010.

[109] S. Leibovich. The structure of vortex breakdown. *Annu. Rev. Fluid Mech.*, 10(1):221–246, 1978.

[110] G. Leonard and J. Stegmaier. Formation of nitric oxide in premixed hydrocarbon flames. *J. Eng. Gas Turbines Power*, 116:542–546, 1993.

[111] B. Lewis and G. von Elbe. *Combustion, Flames, and Explosions of Gases*. Academic Press, 1987.

[112] M. Leyko, F. Nicoud, and T. Poinsot. Comparison of direct and indirect combustion noise mechanisms in a model combustor. 2009.

[113] M. Leyko, S. Moreau, F. Nicoud, and T. Poinsot. Numerical and analytical modelling of entropy noise in a supersonic nozzle with a shock. *J. Sound Vib.*, 330(16):3944–3958, 2011.

[114] T. Lieuwen. *Unsteady combustor physics*. Cambridge University Press, 2012.

[115] T. Lieuwen and V. Yang. *Combustion instabilities in gas turbine engines: operational experience, fundamental mechanisms and modeling*. AIAA, 2005.

[116] T. Lieuwen, V. McDonell, E. Petersen, and D. Santavicca. Fuel flexibility influences on premixed combustor blowout, flashback, autoignition, and stability. *ASME. J. Eng. Gas Turbines Power*, 130(1):011506–011506–10, 2008.

[117] E. N. Lorenz. Deterministic nonperiodic flow. *J. Atmos. Sci.*, 20(2):130–141, 1963.

[118] L. Lovachev. Flame propagation in vortices. *Combust. Flame*, 27:125–127, 1976.

[119] O. Lucca-Negro and T. O'Doherty. Vortex breakdown: a review. *Prog. Energy Combust. Sci.*, 27(4):431 – 481, 2001.

[120] M. Macquisten and A. Dowling. Low-frequency combustion oscillations in a model afterburner. *Combust. Flame*, 94(3):253 – 264, 1993.

[121] L. Magri, J. O'Brien, and M. Ihme. Compositional inhomogeneities as a source of indirect combustion noise. *J. Fluid Mech.*, 799:R4, 2017.

[122] F. E. Marble and S. M. Candel. Acoustic disturbance from gas non-uniformities convected through a nozzle. *J. Sound Vib.*, 55(2):225–243, 1977.

[123] F. D. Mare. *Large eddy simulation of reacting and non-reacting turbulent flows in complex geometries*. PhD thesis, Imperial College London, 2002.

[124] L. D. Mare and W. Jones. LES of turbulent flow past a swept fence. *International Journal of Heat and Fluid Flow*, 24(4):606 – 615, 2003.

[125] C. J. Marek, L. C. Papathakos, and P. W. Verbulecz. Preliminary studies of autoignition and flashback in a premixing-prevaporizing flame tube using jet-a fuel at lean equivalence ratios. Technical Report NASA-TM-X-3526, NASA Lewis Research Center, Cleveland, OH, United States, 1977.

[126] A. D. Margolin and V. P. Karpov. Flame propagation in an eddy combustion chamber. In *SAE Technical Paper*. SAE International, 1974.

[127] N. Marwan, M. C. Romano, M. Thiel, and J. Kurths. Recurrence plots for the analysis of complex systems. *Phys. Rep.*, 738:237–329, 2007.

[128] G. B. Mindlin, X.-J. Hou, H. G. Solari, R. Gilmore, and N. B. Tufillaro. Classification of strange attractors by integers. *Phys. Rev. Lett.*, 64:2350–2353, 1990.

[129] T. Miyano. Time series analysis of complex dynamical behavior contaminated with observational noise. *Int. J. Bifur. Chaos*, 06(11):2031–2045, 1996.

[130] W. H. Moase, M. J. Brear, and C. Manzie. The forced response of chocked nozzles and supersonic diffusers. *J. Fluid Mech.*, 585:281–304, 2007.

[131] A. S. Morgans and I. Duran. Entropy noise: A review of theory, progress and challenges. *Int. J. Spray and Comb. Dyn.*, 8(4):285–298, 2016.

[132] A. S. Morgans, C. S. Goh, and J. A. Dahan. The dissipation and dispersion of entropy waves in combustor thermoacoustics. *J. Fluid Mech.*, 733:1–11, 2013.

[133] Y. Morinishi. Conservation properties of finite difference schemes for incompressible flow. *Center Turbul. Res.*, pages 121–132, 1995.

[134] E. Motheau, F. Nicoud, and T. Poinsot. Mixed acoustic - entropy combustion instabilities in gas turbines. *J. Fluid Mech.*, 749:542–576, 2014.

[135] A. Nauert, P. Petersson, M. Linne, and A. Dreizler. Experimental analysis of flashback in lean premixed swirling flames: conditions close to flashback. *Exp. Fluids*, 43(1):89–100, 2007.

[136] D. R. Noble, Q. Zhang, A. Shareef, J. Tootle, A. Meyers, and T. Lieuwen. Syngas mixture composition effects upon flashback and blowout. *Proc. Combust. Inst.*, 1:357–368, 2006.

[137] J. Oefelein and V. Yang. Comprehensive review of liquid propellant combustion instabilities in f-1 engines. *J. Propul. Power*, 9(5):657–677, 1993.

[138] A. P. Dowling and A. S. Morgans. Feedback control of combustion oscillations. *Annu. Rev. Fluid Mech.*, 37:151–182, 2005.

[139] N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. Shaw. Geometry from a time series. *Phys. Rev. Lett.*, 45:712–716, 1980.

[140] M. C. Paul and M. M. Molla. Investigation of physiological pulsatile flow in a model arterial stenosis using large-eddy and direct numerical simulations. *Appl. Math. Modell.*, 36:4393–4413, 2012.

[141] R. Pavri and G. D. Moore. Gas turbine emissions and control. Technical Report GER-4211, GE Power Systems, 2001.

[142] H. Poincaré. Sur la probleme des trois corps et les équations de la dynamique. *Acta Mathematica*, 13:1–270, 1890.

[143] T. Poinsot. Prediction and control of combustion instabilities in real engines. *Proc. Combust. Inst.*, 36(1):1 – 28, 2017.

[144] T. Poinsot and D. Veynante. *Theoretical and Numerical Combustion*. Edwards, 2005.

[145] W. Polifke, C. Paschereit, and K. Dobbeling. Constructive and destructive interference of acoustic and entropy waves in a premixed combustor with a chocked exit. *Int. J. Acoust. Vib.*, 6(3):135–146, 2001.

[146] A. Rausch, A. Fischer, H. J. Konle, A. Gaertlein, S. Nitsch, K. Knobloch, F. Bake, and I. Röhle. Measurements of density pulsations in the outlet nozzle of a combustion chamber by rayleigh-scattering searching entropy waves. *J. Eng. Gas Turb. Power*, 133: 031601, 2011.

[147] J. W. S. Rayleigh. *The theory of sound*. Number V. 2. Macmillan, 1878.

[148] C. Rhie and W. Chow. Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA J.*, 21(11):1525–1532, 1983.

[149] E. O. Rolland, F. D. Domenico, and S. Hochgreb. Direct and indirect noise generated by entropic and compositional inhomogeneities. *ASME. J. Eng. Gas Turbines Power*, 140 (8):082604, 2018.

[150] V. D. Sarli and A. D. Benedetto. Laminar burning velocity of hydrogen-methane/air premixed flames. *Int. J. Hydrog. Energy*, 32:637–646, 2007.

[151] T. Sattelmayer. Influence of the combustor aerodynamics on combustion instabilities from equivalence ratio fluctuations. *J. Eng. Gas Turbines Power*, 125:11–19, 2003.

[152] T. Sattelmayer, C. Mayer, and J. Sangl. Interaction of flame flashback mechanisms in premixed hydrogen - air swirl flames. *ASME. J. Eng. Gas Turbines Power*, 138(1):011503–011503–12, 2015.

[153] C. Schneider, A. Dreizler, and J. Janicka. Fluid dynamical analysis of atmospheric reacting and isothermal swirling flows. *Flow Turbul. Combust.*, 74(1):103–127, 2005.

[154] T. S. Snyder, T. J. Rosfjord, J. B. McVey, and L. M. Chiappetta. Comparison of liquid fuel /air mixing and nox emissions for a tangential entry nozzle. *ASME. Turbo Expo: Power for Land, Sea, and Air*, 3:94–GT–283, 1994.

[155] Y. Sommerer, D. Galley, T. Poinsot, S. Ducruix, F. Lacas, and D. Veynante. Large eddy simulation and experimental study of flashback and blow-off in a lean partially premixed swirled burner. *J. Turbul.*, 5:N37, 2004.

[156] S. R. Stow, A. P. Dowling, and T. P. Hynes. Reflection of circumferential modes in a choked nozzle. *J. Fluid Mech.*, 467:215–239, 2002.

[157] W. Strahle. Combustion noise. *Prog. Energy Combust. Sci.*, 4(3):157–176, 1978.

[158] S. H. Strogatz. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*. CRC Press, 2018.

[159] N. Syred, M. Abdulsada, A. Griffiths, T. O'Doherty, and P. Bowen. The effect of hydrogen containing fuel blends upon flashback in swirl burners. *Appl. Energy*, 89(1):106 – 110, 2012.

[160] N. Syred, A. Giles, J. Lewis, M. Abdulsada, A. V. Medina, R. Marsh, P. Bowen, and A. Griffiths. Effect of inlet and outlet configurations on blow-off and flashback with premixed combustion for methane and a high hydrogen content fuel in a generic swirl burner. *Appl. Energy*, 116:288 – 296, 2014.

[161] S. Taamallah, K. Vogiatzaki, F. M. Alzahrani, E. M. A. Mokheimer, M. A. Habib, and A. F. Ghoniem. Fuel flexibility, stability and emissions in premixed hydrogen-rich gas turbine combustion: Technology, fundamentals, and numerical simulations. *Appl. Energy*, 154(C):1020–1047, 2015.

[162] F. Takens. Detecting strange attractors in turbulence. *Lect. Notes Math.*, 898:366, 1981.

[163] D. Thibaut and S. Candel. Numerical study of unsteady turbulent premixed combustion: application to flashback simulation. *Combust. Flame*, 113(1):53–65, 1998.

[164] O. Tuncer, S. Acharya, and J. Uhm. Hydrogen-enriched confined methane flame behavior and flashback modelling. 2006.

[165] O. Tuncer, S. Acharya, and J. Uhm. Dynamics, NOx and flashback characteristics of confined premixed hydrogen-enriched methane flames. *Int. J. Hydrogen Energy*, 34:496–506, 2009.

[166] S. Turns. *An Introduction to Combustion: Concepts and Applications*. McGraw-Hill, 2012.

[167] A. Umemura and S. Takamori. Wave nature in vortex-bursting initiation. *Proc. Combust. Inst.*, 28(2):1941 – 1948, 2000.

[168] L. Vaneveld, K. Hom, and A. K. Oppenheim. Secondary effects in combustion instabilities leading to flashback. *AIAA J.*, 22(1):81–82, 1984.

[169] H. Vorst. Bi-cgstab: a first and smoothly converging variant of bi-cg for the solution of non-symmetric linear systems. *SIAM J. Sci. Stat.Comput.*, 13(2):631–644, 1992.

[170] J. Wang, Z. Huang, C. Tang, H. Miao, and X. Wang. Numerical study of the effect of hydrogen addition on methane–air mixtures combustion. *Int. J. Hydrog. Energy*, 34(2): 1084 – 1096, 2009.

[171] D. Wassmer, B. Schuermans, C. O. Paschereit, and J. P. Moeck. Measurement and modelling of the generation and the transport of entropy waves in a model gas turbine combustor. *Int. J. Spray Combust.*, 9(4):299–309, 2017.

[172] R. Wayland, D. Bromley, D. Pickett, and A. Passamante. Recognising determinism in a time series. *Phys. Rev. Lett.*, 70:580–582, 1993.

[173] C. L. Webber and J. P. Zbilut. Dynamical assessment of physiological systems and states using recurrence plot strategies. *J. Appl. Physiol.*, 76(2):965–973, 1994.

[174] J. E. F. Williams and M. Howe. The generation of sound by density inhomogeneities in low mach number nozzle flows. *J. Fluid Mech.*, 70:605–622, 1975.

[175] K. Wohl. Quenching, flash-back, blow-off-theory and experiment. *Symp. (Int.) Combust.*, 4(1):68–89, 1953.

[176] M. M. Zhu, A. P. Dowling, and K. C. Bray. Self-excited oscillations in combustors with spray atomizers. *J. Eng. Gas Turbines Power*, 123(4):779–786, 2000.

[177] O. Zikanov. *Essential computational fluid dynamics*. John Wiley & Sons, 2010.

[178] E. E. Zukoski and J. M. Auerbach. Experiments concerning the response of supersonic nozzles to fluctuating inlet conditions. *ASME. J. Eng. Power*, 98(1):60–64, 1976.

Appendix A

Codes developed for nonlinear time series analysis and modelling

A.1 Main programs

The codes that follow are written in the MATLAB programming language. The codes in section A.1.1, process a time series that is supplied by the user using the nonlinear time series analysis methods in chapter 2. These codes have been used to process the pressure time series from the flashback experiments (not part of this work) and to generate the results in the flashback study, in chapter 3. The code in section A.1.2, reads temperature field data that are supplied by the user in data files and processes them to generate the case specific model of an advecting entropy wave in turbulent channel flow. This code has been developed based on the novel methodology presented in chapter 4 and used to generate the results in the entropy wave study, in section 4.3.

A.1.1 Nonlinear time series analysis

Generating a plot of the translation error of a system's trajectory in phase space as a function of time from the time series of a measured system variable

The following program calculates the translation error of a dynamical system's trajectory in phase space by analysing a time series of a measured variable from the system. The program prompts the user for a time series s and its sampling frequency fs. Further, it requests the parameters needed for the translation error calculation. These are, the number of random centre points M, the number of sets of M which is Q, the number of nearest neighbours k, the width of the running window measured in data points, and the displacement of the running window also measured in data points. For example, to use a running window with 50% overlap, the width of the window would be w data points and the window displacement would be $w/2$ data points. The program takes sections of the time series s one window at a time and calculates the coordinates of the phase points that reconstruct the section of the trajectory of the system in

phase space for the interval captured by the window. This calculation is done using the method of time delay embedding. The program calls the functions AMI and FNN in sections A.2.1 and A.2.2. These functions return the time delay T and the embedding dimension d needed for the embedding. The time delay T and embedding dimension d only need to be calculated once. Having calculated the coordinates of the phase points, the translation error is calculated and stored in array *Etrans*. The window is displaced and the calculations are repeated for the next window. Finally, the program generates a plots of the supplied time series and the translation error as a function of time.

```
% inputs
s = input('TimeSeries: ');
fs = input('SamplingFrequency: ');
M = input('NumberOfRandomCentres: ');
Q = input('SetsOfRCs: ');
k = input('NoNearestNeighbors: ');
L_win = input('WindowLength: ');
WindowDisplacement = input('WindowDisplacement: ');
% if time series is a column vector
% then convert it to a row vector
[rows,~] = size(s);
if rows ~=1; s = s'; end
% length of input time series
L_tms = length(TimeSeries);
if L_tms<2e3
    ss = s;
else
    ss = s(1:2e3);
end
% find delay T for embedding
T = AMI(ss,81);
% find dimension d for embedding
d = FNN(ss,T);
% initialize window trailing edge (TE) and leading edge (LE)
window_TE = 1;
window_LE = L_win;
% initialize a window counter
window = 1;
% for each window calculate translation error E_trans
while (window_LE < L_tms) || (window_LE == L_tms)
    % generate matrix y. Each column of y contains the
    % coordinates of a phase point in phase space
    for T_coeff = 0:(d-1)
        s_1 = window_TE + T_coeff*T;
        s_n = window_LE-(d-1)*T + T_coeff*T;
        y( (T_coeff+1),: ) = s(s_1:s_n);
    end
    % total number of phase points in phase space
    l = length(y);
    % for every window:
```

```
    % 1. select M random centre points.
    % 2. calc translation error at each random centre point.
    % 3. calc median of the M translation errors from step 2.
    % 4. repeat steps 1-3 Q times.
    % 5. calc mean of the Q translation errors. This is the
    % translation error for the window.
    % 6. repeat steps 1-5 for all other windows.
    for set = 1:Q
        % the M random centre points.
        RC_points = randi(1,[1,M]);
        % calc translation error e_trans at each RC point
        % initialize a position counter for the arrays.
        count = 1;
        for RC_point = RC_points
            % find k nearest neighbors to RC_point
            % eucledian distance between RC_point and
            % every other point on trajectory
            E=sqrt(sum((y(:,RC_point)*ones(1,1)-y(:,:)).^2));
            % ignore the zero Euclidean distance which is
            % the ditance between RC_point and itself
            E(RC_point) = NaN;
            % nearest neighbor to RC_point is y(:,index)
            [~, index] = min(E);
            % find the other k-1 nearest neighbors
            for n = 1:k
                y_NN(1,n) = index;
                E(index) = NaN;
                [~, index] = min(E);
            end
            % column one of Y is RC_point and the other k
            % columns are the k nearest neighbors
            Y = cat(2,RC_point,y_NN);
            % translation vectors
            for i = 1:(k+1)
                % if phase point i (column i in Y) is the
                % last point on the trajectory ignore it
                % (it does not have an image!)
                if Y(i)~=1
                    V(:,i) = y(:,Y(i)+1) - y(:,Y(i));
                end
            end
            % average translation vector
            V_average = ( sum(V,2) ./ (k+1) );
            % translation error at RC_point
            e_trans=sum(sum((V-V_average*ones(1,k+1)).^2)...
            ./sum(V_average.^2))/(k+1);
            % store the translation error
            e_trans_array(1,count) = e_trans;
            count = count + 1;
        end % next RC point
        % translation error for the set
```

```
        e_trans_median(1,set) = median(e_trans_array);
    end % next set
    % translation error for the current window
    E_trans(1,window) = mean(e_trans_median);
    % trans. error for the current window is plotted at
    %((time window ends)-(time window starts))/2
    t_for_E_trans(1,window) = (1/fs)*(window_TE + L_win/2);
    % next window
    window = window + 1;
    % next window trailing edge (TE) and leading edge (LE)
    window_TE = window_TE + WindowDisplacement;
    % window_TE = window_LE - floor( overlap*L_win );
    window_LE = window_TE + L_win;
    %
    clear y;
end
% smoothing - mooving average
E_trans = smooth(E_trans);
% generate plots
figure;
% time series plot
t_for_time_series = (1/fs)*( 1:L_tms );
p = s;
subplot(2,1,1)
plot(t_for_time_series,p)
xlabel('t (s)','fontsize',12)
ylabel('p\prime','fontsize',12)
% translation error plot
subplot(2,1,2)
plot(t_for_E_trans,E_trans)
xlabel('t (s)','fontsize',12)
ylabel('e_{trans}','fontsize',12)
% mean plus/minus 1std
leg = length(E_trans);
hold on;
plot(t_for_E_trans,mean(E_trans)*ones(1,leg));
plot(t_for_E_trans,(mean(E_trans)+std(E_trans))*ones(1,leg));
plot(t_for_E_trans,(mean(E_trans)-std(E_trans))*ones(1,leg));
hold off;
```

Generating recurrence plots and recurrence quantification measures as a function of time for a system's trajectory in phase space from a time series of a measured system variable

The following program generates the recurrence matrix and the recurrence plot for the trajectory of a system that is reconstructed in phase space from a time series supplied by the user using the time delay embedding method. The program prompts the user for a time series s and its sampling frequency fs. Further, it requests the size of the neighbourhood that will be used to check for recurrences of the trajectory in phase space. The size of the neighbourhood is an L_2

norm or Euclidean norm. The user supplies the size of the neighbourhood as a percentage of the maximum attractor diameter (range is 0-1). The maximum attractor diameter is the maximum distance between any two phase points along the trajectory. Finally, the user supplies the width of the running window measured in data points, and the displacement of the running window also measured in data points. For example, to use a running window with 50% overlap, the width of the window would be w data points and the window displacement would be $w/2$ data points. The program takes sections of the time series s one window at a time and calculates the coordinates of the phase points that reconstruct the section of the trajectory of the system in phase space for the interval captured by the window. This calculation is done using the method of time delay embedding. The program calls the functions AMI and FNN in sections A.2.1 and A.2.2. These functions return the time delay T and the embedding dimension d needed for the embedding. The time delay T and embedding dimension d only need to be calculated once. Having calculated the coordinates of the phase points, the neighbourhood of every phase point on the trajectory is searched and the recurrences are recorded in the recurrence matrix. The recurrence matrix is binary, where a one represents a recurrence and a zero represents no recurrence. The window is displaced and the process is repeated for the next window. The program generates the recurrence plot of every window and the plots of the recurrence quantification measures (RR, DET, and DIV) as a function of time.

```
% clear command window
clc
% Record the time the simulation starts
t_1 = clock;
% Inputs
ts = input('Time series: ');
clearvars -except ts t_1
fs = input('Sampling frequency: ');
percent_dA=input('Neighborhood (% of attractor diameter): ');
L_win = input('WindowLength: ');
WindowDisplacement = input('WindowDisplacement: ');
L_win = L_win - 1;
% If time series is a column vector convert to row vector
[rows,~] = size(ts);
if rows ~=1; ts = ts'; end
% Number of data points in time series 'N_tms'
L_tms = size(ts,2);
% find T and d for embedding
if L_tms<2e3
    ss = ts;
else
    ss = ts(1:2e3);
end
% find delay T for embedding
T = AMI(ss,81);
% find dimension d for embedding
D = FNN(ss,T) + 1;
```

```
% sampling time
t_s = 1/fs;
% Initialize window trailing edge (TE) and leading edge (LE)
window_TE = 1;
window_LE = window_TE + L_win;
% Initialize a window counter
window = 1;
% for each window calculate the RQA measures
while (window_LE < L_tms) || (window_LE == L_tms)
    point0 = window_TE;
    dotCount = 1;
    % Generate matrix y.
    % Each column of 'y' contains the coordinates
    % of a phase point in phase space
    for T_coeff = 0:(D-1)
        y((T_coeff+1),:) = ts((window_TE+T_coeff*T):...
            (window_LE-(D-1)*T+T_coeff*T));
    end
    % Number of phase points in phase space 'N_ps'
    N_ps = size(y,2);
    % Diameter of attractor
    for j = 1:N_ps
        % Eucledian distance
        E = sqrt(sum((y(:,j)*ones(1,N_ps)-y(:,1:N_ps)).^2));
        E_max(1,j) = max(E);
    end
    d_A = max(E_max);
    % Define recurrence threshold 'epsilon'
    epsilon = percent_dA*d_A;
    % Generate histogram of diagonal line lengths P_l
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Minimum diagonal line length is taken to be of the
    % order of the autocorrelation time
    l_min = 2;
    % Initialize
    l = 0;
    P_l = zeros(1,N_ps);
    for tau = 0 : ( N_ps - 1 )
        for i = 1 : ( N_ps-tau ) % for each phase point
            j = i + tau; % co-ordinate pair
            % Calculate the Euclidean norm between i and j
            E = sqrt( sum( ( y(:,i) - y(:,j) ).^2 ) );
            if ( E < epsilon ) || ( E == epsilon )
                if ( tau == 0 ) % if black dot is ON THE LOI
                    % save
                    dotXYs(dotCount,:) = ...
                        [(i-1+point0)*t_s,(j-1+point0)*t_s];
                else % if the dot is NOT ON THE LOI
                    % save
                    dotXYs(dotCount,:) = ...
                        [(i-1+point0)*t_s,(j-1+point0)*t_s];
```

```
                    % and
                    dotCount = dotCount + 1;
                    % mirror image
                    dotXYs(dotCount,:) = ...
                        [(j-1+point0)*t_s,(i-1+point0)*t_s];
                end
                dotCount = dotCount + 1;
                l = l + 1;
            elseif ( l ~= 0 )
                P_l(l) = P_l(l) + 1;
                l = 0;
            end
            if ( i == N_ps - tau ) && ( l ~= 0 )
                P_l(l) = P_l(l) + 1;
                l = 0;
            end
        end
    end
    P_l = [2 .* P_l( 1 : N_ps-1 ), P_l(N_ps)];

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % CALCULATE RQA MEASURES
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Calculate the Recurrence Rate RR
    RR = sum( ( 1 : N_ps ) .* P_l ) / N_ps^2;
    % Calculate RQA measures based on diagonal lines
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Calculate the determinism (excluding LOI)
    sumNumerator = sum((l_min:N_ps-1).* P_l(l_min:N_ps-1));
    sumDenominator = sum((1:N_ps-1).* P_l(1:N_ps-1));
    DET = sumNumerator/sumDenominator;
    % Calculate the average diagonal line length (ex. LOI)
    sumNumerator = sum((l_min:N_ps-1).* P_l(l_min:N_ps-1));
    sumDenominator = sum(  P_l( l_min : N_ps-1 ) );
    L = sumNumerator/sumDenominator;
    % Length of longest diagonal and its reciprocal (ex. LOI)
    l = N_ps-1;
    while (P_l(l)==0) && (( l > l_min)||(l==l_min))
        l = l - 1;
    end
    L_max = l;
    DIV = 1/L_max;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % SAVE RQA MEASURES FOR WINDOW
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Store recurrence rate 'RR'
    RR_array(1,window) = RR;
    % Store determinism 'DET'
    DET_array(1,window) = DET;
    % Store length of longest diagonal and its reciprocal
    DIV_array(1,window) = DIV;
```

```
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % PLOT RP FOR WINDOW
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    figure('name',['RP for window=',num2str(window)])
    scatter(dotXYs(:,1),dotXYs(:,2),'.','sizedata',10);
    xlabel('time (s)'); ylabel('time (s)');
    axis square; box on;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Print progress to command window
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    clc;
    Progress = window_LE/L_tms*100;
    fprintf('\n\nProgress: %.2f%%',Progress);
    t(1,window) = (1/fs)*(window_TE + L_win/2);
    % Next window
    window = window + 1;
    % Next window trailing edge (TE) and leading edge (LE)
    window_TE = window_TE + WindowDisplacement;
    % window_TE = window_LE - floor( overlap*L_win );
    window_LE = window_TE + L_win;
    clear y dotCount dotXYs;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PLOT RQA MEASURES AS A FUNCTION OF TIME
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure('name','Recurrence rate (RR)');
plot(t,RR_array); xlabel('t (s)'); ylabel('RR');
figure('name','Determinism (DET)');
plot(t,DET_array); xlabel('t (s)'); ylabel('DET');
figure('name','Divergence (DIV)');
plot(t,DIV_array); xlabel('t (s)'); ylabel('DIV');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Clear command window
clc;
% Record the time the simulation ends
t_2 = clock;
% Execution time
t_exe = etime(t_2,t_1);
mins = floor(t_exe/60);
secs = round(t_exe - mins*60);
fprintf('The calculation completed in: ');
fprintf('%imin and %isec.\n\n',mins,secs);
```

A.1.2 Low order modelling

The following program processes the temperature field data (in this case DNS data), which are read from data files and generates plots that show the actual and model generated shape and amplitude of the two-dimensional entropy wave during the advection. Furthermore, the

program generates plots of the equation coefficients and Akaike weights as a function of the wall-normal coordinate, and plots of the model performance relative to the actual behaviour of the wave (deviation and correlation plots). The program first reads the temperature field data from data files. It then searches for the maximum temperature in each streamwise cross-section of the flow and saves its magnitude and streamwise position. These give the (DNS) amplitude and shape of the entropy wave over the channel cross-section and are stored in the arrays *T_dns* and *x_dns*, respectively. The rows of these arrays contain the (DNS) position and (DNS) amplitude with respect to the wall-normal coordinate and the columns contain the (DNS) position and (DNS) amplitude with respect to time. The program calls the function ODEpar in section A.2.3 to calculate the coefficients of the model equations that are case specific. Once the coefficients are returned, the main program moves on to solve the model equations numerically with the MATLAB function ode113. This ODE solver requires that the equations be defined in a function. Therefore, the function ODEeqn in section A.2.4 defines the model equations and is called by the MATLAB function ode113. The amplitude and position of the entropy wave from the solution of the model equations are stored in the arrays *T_lom* and *x_lom*, respectively. The deviation of the position and amplitude of the wave from the model relative to the actual values (in this case from DNS) is calculated. Furthermore, the correlation between the actual and model derived wave shape and amplitude variation over the cross-section is calculated. Finally, the various plots are generated.

```
% clear command window and workspace
clc; clear;
t_start = clock;
fprintf('\n*plot_advection_v2.m is running...\n\n');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SIMULATION PARAMETERS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Lx = 0.0754; % channel length
delta = 0.006; % channel half-width
Ubulk = 102; % bulk velocity
Tmean = 1500; % mean temperature
% print simulation parameters to command window
fprintf('!! MAKE SURE THESE PARAMETERS ARE CORRECT !!\n');
fprintf('simulation parameters are set to: ');
fprintf('Lx/delta/Tmean/Ubulk = ');
fprintf('%1.4fm/%5.3fm/%iK/%6.2fmps\n',Lx,delta,Tmean,Ubulk);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% INPUTS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
os = input('lin or win: ','s'); % operating system
filepath = input('filepath: ','s'); % filepath
% nodes
NX = input('nodes in streamwise direction: ');
NY = input('nodes in wall-normal direction: ');
% total DNS time steps
```

```
total_dns_iter = input('total DNS time steps: ');
% time instants to plot data (in iters of DNS time step)
dns_iter_array = ...
input('time steps to plot [first:step:last, min=6!!]: ');
while (length(dns_iter_array) < 6) && ...
      (isempty(dns_iter_array)~=1)
d = dialog('Name','Warning');
txt = uicontrol('Parent',d,'Style','text',...
    'String','There should be a minimum of 6 time steps!!');
btn = uicontrol('Parent',d,'String','Close',...
                                    'Callback','delete(gcf)');
dns_iter_array = ...
input('Time steps to plot [first:step:last, min=6!!]: ');
end
% build model using partial data (partial data percentage)
pdp = input('Partial data percentage (max. 3): ');
% long exposure on or off
exposure = input('Long exposure (true/false): ');
fprintf('\n'); % move cursor to next line
% partial data sets
all = length(dns_iter_array);
points_for_fits = round(all*pdp);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% GET PARAMETERS ppa and h_coeff FROM DATA FILENAME
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% pulse percentage amplitude (% of Tmean)
ppa = str2double(dataset(4:5))/100;
% heat tranfer convection coefficient
h_coeff = str2double(dataset(9:11));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MAIN PROGRAM
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ITERATIONS - TIME STEPS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
first_iter = 1;
last_iter = length(dns_iter_array);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% DATA ACQUISITION - DNS TIME ARRAY
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% add path to data filename
if strcmp(os,'lin')==1
    filename = [filepath,'/t'];
else
    filename = [filepath,'\t'];
end
%
fileID = fopen(filename); % open file
t_dns = fscanf(fileID,'%f',[1,total_dns_iter]); % read file
fclose(fileID); % close file
t_dns=t_dns'; % transpose data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% MATLAB TIME ARRAY - see dns_iter_array above in INPUTS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% time instants
for iter = first_iter : last_iter
    t(iter) = t_dns(dns_iter_array(iter));
end
t = t.*Ubulk./Lx;     % NON-DIMENSIONAL TIME!!!
Dt = diff(t); % matlab time step (downsampled)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% DATA ACQUISITION - DNS TEMP FIELD
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% initial condition (x_max and T_max at t=0)
for iter = first_iter:last_iter
    % clear dns data variables before next iteration
    clear x_field y_field z_field
    % dns iteration
    dns_iter = dns_iter_array(iter);
    % data file name
    filename = ['T_',num2str(dns_iter,'%04u')];
    % add path to data file name
    if strcmp(os,'lin')==1
        filename = [filepath,filename];
    else
        filename = [filepath,filename];
    end
    %
    fileID = fopen(filename); % open file
    field = fscanf(fileID,'%f',[3,NX*NY]); % read file
    fclose(fileID); % close file
    field=field'; % transpose data
    % non-dimensional y
    if iter==1
        for j = 1:NY
            y_dns(j) = field(j,2)/delta;
        end
    end
    % non-dimensional x and T
    for i = 1:NX
        % non-dimensional x
        x_field(i) = field((i-1)*NY+1,1)/delta;
        % non-dimensional T
        for j =  1:NY
            if (field(j+(i-1)*NY,3)>Tmean)
                T_field(j,i) = ...
                (field(j+(i-1)*NY,3)-Tmean)./(Tmean);
            else
                T_field(j,i) = 0; % cap values at zero
            end
        end
    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
    % T AND X OF MAX TEMP. IN EACH STREAMWISE SECTION
    %%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% model build-up
for model = 1 : length(points_for_fits)
    jnonzero = 1;
    for j = 1:NY
        if j_ignore(j) ~= 1
            y = y_dns(j);
            % solve system of eqns
            [tt,out] = ode113(...
                        @(c,y)ODEeqn(c,y,...
                                    theta1(model,jnonzero),...
                                    theta2(model,jnonzero),...
                    bestfit(model,jnonzero)),...
                            t,[x_dns(1,j),T_dns(1,j)]);
            % store model data in xmodel and Tmodel
            A_i = (model-1)*length(tt)+1;
            Z_i = (model-1)*length(tt)+length(tt);
            %
            x_lom_eff(A_i:Z_i,jnonzero)=out(1:length(tt),1);
            y_lom_eff(A_i:Z_i,jnonzero)=y*ones(length(tt),1);
            T_lom_eff(A_i:Z_i,jnonzero)=out(1:length(tt),2);
            jnonzero = jnonzero + 1;
        end
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% GENERATE FRAMES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for iter = first_iter:last_iter % time steps
    % clear field variables
    clear x_max y_max z_max
    %
    dns_iter = dns_iter_array(iter);
    % NEED EFFECTIVE DNS ARRAYS.
    % ALREADY HAVE LOM EFFECTIVE ARRAYS.
    % EFFECTIVE ARRAYS DO NOT INCLUDE j_ignore(j)==1 ELEMENTS
    % j_ignore(j)==1 ARE THE STREAMWISE SECTIONS THAT CONTAIN
    % L.T. 6 NONZERO STATES WHICH IS THE MIN REQUIERED TO
    % MAKE FITS AND CREATE THE LOM FOR THAT Y-LAYER.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    jnonzero = 1;
    for j = 1:NY
        if j_ignore(j) ~= 1
            x_dns_eff(1,jnonzero) = x_dns(iter,j);
            y_dns_eff(1,jnonzero) = y_dns(j);
            T_dns_eff(1,jnonzero) = T_dns(iter,j);
            jnonzero = jnonzero + 1;
        end
    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % LOM PERFORMANCE STATISTICS
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PREPARE ARRAYS FOR PLOTTING - IGNORE ZERO VALUES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% DNS
count = 1;
clear x_dns_plot y_dns_plot T_dns_plot
for j = 1:NY
    if j_ignore(j) ~= 1
        % check if amplitude has reached zero
        % if true then ignore y-slice
        if T_dns(iter,j) > 10e-3
            x_dns_plot(1,count) = x_dns(iter,j);
            y_dns_plot(1,count) = y_dns(j);
            T_dns_plot(1,count) = T_dns(iter,j);
            count = count + 1;
        end
    end
end
% LOM
for model = 1:length(points_for_fits)
    count = 1; jnonzero=1;
    clear x_lom_plot y_lom_plot T_lom_plot
    for j = 1:NY
        if j_ignore(j) ~= 1
            % check if amplitude has reached zero
            % if true then ignore this y-slice
            if T_dns(iter,j) > 10e-3
                %
                        A = (model-1)*length(tt)+iter;
                %
                x_lom_plot(count)=x_lom_eff(A,jnonzero);
                y_lom_plot(count)=y_lom_eff(A,jnonzero);
                T_lom_plot(count)=T_lom_eff(A,jnonzero);
                count = count + 1;
            end
            jnonzero = jnonzero + 1;
        end
    end
end
% SMOOTH LOM DATA
[i_end,j_end] = size(x_lom_plot);
for i = 1 : i_end
    x_lom_plot(i,:) = ...
    smoothdata(x_lom_plot(i,:),'SmoothingFactor',0.5);
end
for j = 1 : length(x_dns_plot)
    x_dns_plot_full(iter,j) = x_dns_plot(j);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% CROSS SECTIONAL AVG
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
281     dy = y_dns(2:end)-y_dns(1:end-1);
282     fraction = ( 1/(y_dns_plot(end)-y_dns_plot(1)) );
283     for model = 1 : length(points_for_fits)
284         A = (model-1)*length(tt)+iter;
285         % error in mean position
286         count=1;xavg_dns=0; xavg_lom=0;
287         for j = 1 : NY-1
288             if (j_ignore(j)~=1)&&(count<length(y_lom_plot))
289                 xavg_dns = xavg_dns + fraction* ...
290                 (x_dns_plot(count) + x_dns_plot(count+1))/...
291                             2*dy(j);
292                 xavg_lom = xavg_lom + fraction* ...
293                 (x_lom_plot(count) + x_lom_plot(count+1))/...
294                             2*dy(j);
295                 count = count + 1;
296             end
297         end
298         xout_dns(iter) = xavg_dns;
299         xout_lom(iter) = xavg_lom;
300         epsilon_x(A) = (xavg_lom-xavg_dns)/xavg_dns*100;
301         % error in mean amplitude
302         count=1;
303         Tavg_dns=0; Tavg_lom=0;
304         for j = 1 : NY-1
305             if (j_ignore(j)~=1)&&(count<length(y_lom_plot))
306                 Tavg_dns = Tavg_dns + fraction* ...
307                 (T_dns_plot(count) + T_dns_plot(count+1))/...
308                             2*dy(j);
309                 Tavg_lom = Tavg_lom + fraction* ...
310                             (T_lom_plot(count) + ...
311                              T_lom_plot(count+1))/2*dy(j);
312                 count = count + 1;
313             end
314         end
315         Tout_dns(iter) = Tavg_dns;
316         Tout_lom(iter) = Tavg_lom;
317         epsilon_T(A) = (Tavg_lom-Tavg_dns)/Tavg_dns*100;
318         % correlation of position
319         count=1;
320         avg_res_xdns=0; avg_res_xlom=0; avg_res_prod=0;
321         for j = 1 : NY-1
322             if (j_ignore(j)~=1)&&(count<length(x_lom_plot))
323                 avg_res_xdns = avg_res_xdns + fraction* ...
324                             ((x_dns_plot(count) - ...
325                                         xavg_dns)^2)*dy(j);
326                 avg_res_xlom = avg_res_xlom + fraction* ...
327                             ((x_lom_plot(count) - ...
328                                         xavg_lom)^2)*dy(j);
329                 avg_res_prod = avg_res_prod + fraction* ...
330                             ((x_dns_plot(count) - ...
331                                     xavg_dns))* ...
```

```
                                ((x_lom_plot(count) - ...
                                        xavg_lom))*dy(j);
                    count = count + 1;
                end
            end
            stdev_xdns = sqrt(avg_res_xdns);
            stdev_xlom = sqrt(avg_res_xlom);
            rho_xx(A) = avg_res_prod/(stdev_xdns*stdev_xlom);
            % correlation of amplitude
            count=1;
            avg_res_Tdns=0; avg_res_Tlom=0; avg_res_prod=0;
            for j = 1 : NY-1
                if (j_ignore(j)~=1)&&(count<length(T_lom_plot))
                    avg_res_Tdns = avg_res_Tdns + fraction* ...
                                ((T_dns_plot(count) - ...
                                        Tavg_dns)^2)*dy(j);
                    avg_res_Tlom = avg_res_Tlom + fraction* ...
                                ((T_lom_plot(count) - ...
                                        Tavg_lom)^2)*dy(j);
                    avg_res_prod = avg_res_prod + fraction* ...
                                ((T_dns_plot(count) - ...
                                        Tavg_dns))* ...
                                ((T_lom_plot(count) - ...
                                        Tavg_lom))*dy(j);
                    count = count + 1;
                end
            end
            stdev_Tdns = sqrt(avg_res_Tdns);
            stdev_Tlom = sqrt(avg_res_Tlom);
            rho_TT(A) = avg_res_prod/(stdev_Tdns*stdev_Tlom);
        end
        %%%%
```

```
383         xlabel('$\bf y/\delta$',...
384                'interpreter','latex','fontsize',fontsize);
385         ylabel('$$\bf \Delta T/\overline{T}$$',...
386                'interpreter','latex','fontsize',fontsize);
387         % axis limits
388         axis([-1 1 0 ppa]);
389     end
390     figure(f1)
391     if exposure; hold on; end
392     plot(a1,y_dns_plot,T_dns_plot,'linestyle','none',...
393             'marker','s','markersize',4,'markerfacecolor',...
394        color_face,'markeredgecolor',color_dns);
395     for model = 1 : length(points_for_fits)
396         plot(a1,y_lom_plot,T_lom_plot,'linestyle','-',...
397              'color',color_lom(model,:),...
398              'linewidth',linewidth);
399     end
400     hold off;
401     % shape in xy section
402     if iter==1
403         f2 = figure;
404         set(f2,'units','normal',...
405             'position',f1_pos+[0.05 -0.1 0 0])
406         f2_pos = get(f2,'position');
407         a2 = axes('fontsize',fontsize-2);
408         grid on; box on; axis square;
409         % axis labels
410         xlabel('$\bf x/\delta$',...
411                'interpreter','latex','fontsize',fontsize);
412         ylabel('$\bf y\delta$',...
413                'interpreter','latex','fontsize',fontsize);
414         % axis limits
415         axis([0 Lx/delta -1 1]);
416         % ticks
417         xlbls=['0\pi';'2\pi';'4\pi'];
418         xtcks=[0,2*pi,4*pi];
419         set(gca,'xtick',xtcks,'xticklabel',xlbls);
420     end
421     figure(f2)
422     if exposure; hold on; end
423         plot(a2,x_dns_plot,y_dns_plot,...
424         'linestyle','none','marker','s','markersize',4,...
425         'markerfacecolor',color_face,...
426         'markeredgecolor',color_dns);
427     for model = 1 : length(points_for_fits)
428         plot(a2,x_lom_plot,y_lom_plot,...
429         'linestyle','-','color',color_lom(model,:),...
430         'linewidth',linewidth);
431     end
432     hold off;
433 end
```

```
% akaike weights
f3 = figure;
set(f3,'units','normal','position',f2_pos+[0.05 -0.1 0 0])
f3_pos = get(f3,'position');
for fitting_no = 1:3
    a3(fitting_no) = subplot(1,3,fitting_no);
    set(a3(fitting_no),'fontsize',fontsize-2);
    grid on; box on; axis square;
    % axis limits
    axis([-1 1 0 1]);
    % axis labels
    title(['fitting ',num2str(fitting_no,'%i')],...
        'interpreter','latex','fontsize',fontsize);
    xlabel('$\bf y/\delta$',...
        'interpreter','latex','fontsize',fontsize);
    ylabel('$$\bf AIC \hspace{1.5mm} weights$$',...
        'interpreter','latex','fontsize',fontsize);
    hold on;
    for model = 1 : length(points_for_fits)
        rows = 1:length(y_eff);
        columns = (model-1)*3+fitting_no;
        bar(a3(fitting_no),y_eff,...
            akaike_weights(rows,columns)',...
            'barwidth',1,'edgecolor','none',...
            'facecolor',color_lom_data(model,:));
    end
    hold off;
end
% theta1(y)
f4 = figure;
set(f4,'units','normal',...
            'position',f3_pos+[0.05 -0.1 0 0])
f4_pos = get(f4,'position');
a4 = axes('fontsize',fontsize-2);
grid on; box on; axis square;
% axis labels
xlabel('$\bf y/\delta$',...
    'interpreter','latex','fontsize',fontsize);
ylabel('$\bf \theta_1$',...
    'interpreter','latex','fontsize',fontsize);
hold on;
for model = 1 : length(points_for_fits)
    plot(a4,y_eff,theta1(model,:),...
        'linestyle','none','marker','s',...
        'markersize',4,'markerfacecolor','none',...
        'markeredgecolor',color_lom_data(model,:));
end
hold off;
% theta2(y)
f5 = figure;
set(f5,'units','normal','position',f4_pos+[0.05 -0.1 0 0])
```

```
f5_pos = get(f5,'position');
a5 = axes;
set(a5,'fontsize',fontsize-2);
grid on; box on; axis square;
% axis labels
xlabel('$\bf y/\delta$',...
    'interpreter','latex','fontsize',fontsize);
ylabel('$\bf \theta_2$',...
    'interpreter','latex','fontsize',fontsize);
hold on;
for model = 1 : length(points_for_fits)
    plot(a5,y_eff,-theta2(model,:),...
        'linestyle','none','marker','s',...
        'markersize',4,'markerfacecolor','none',...
        'markeredgecolor',color_lom_data(model,:));
end
hold off;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% END OF GENERATE FRAME FOR EACH TIME INSTANT
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PERFORMANCE STATISTICS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for model = 1 : length(points_for_fits)
        %
        A = (model-1)*length(tt)+1;
        Z = (model-1)*length(tt)+length(tt);
        % new figure
        hf3(model) = figure('name',...
                            ['model ',num2str(model,'%i')],...
                            'WindowStyle','normal');
        set(hf3(model),'units','normal',...
            'position',f5_pos+[0.05*model -0.1*model 0 0])
        % average deviation
        subplot(1,2,1)
        axis square; grid on; box on;
        set(gca,'fontsize',fontsize);
        xlabel('$\bf \hat{t}$','interpreter','latex',...
               'fontsize',fontsize);
        ylabel('$\bf \epsilon_{\hat{\Phi}} (\%)$',...
               'interpreter','latex','fontsize',fontsize);
        hold on;
        plot((t-t(1)).*(1e-3*Ubulk/Lx),epsilon_x(A:Z),...
            'sk','markerfacecolor','none');
        plot((t-t(1)).*(1e-3*Ubulk/Lx),epsilon_T(A:Z),...
            'sk','markerfacecolor','k');
        hold off;
        % correlation over cross-section
        subplot(1,2,2)
        axis square; grid on; box on;
        set(gca,'fontsize',fontsize);
        xlabel('$\bf \hat{t}$','interpreter','latex',...
```

```
                'fontsize',fontsize);
        ylabel('$\bf \rho_{\hat{\Phi}}$',...
                'interpreter','latex','fontsize',fontsize);
        hold on;
        plot((t-t(1)).*(1e-3*Ubulk/Lx),rho_xx(A:Z),...
            'sk','markerfacecolor','none');
        plot((t-t(1)).*(1e-3*Ubulk/Lx),rho_TT(A:Z),...
            'sk','markerfacecolor','k');
        axis([-inf inf 0 1])
        hold off;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
t_end = clock; % run end time
% code run time
dt_run = etime(t_end,t_start);
hours = dt_run/60/60;
mins = (hours-floor(hours))*60;
secs = (mins-floor(mins))*60;
fprintf('*run finished...in %02i:%02i:%02i\n\n',...
floor(hours),floor(mins),floor(secs));
```

A.2 Functions

A.2.1 Function AMI

The following function determines the time delay T for the time delay embedding of a time series using the method of average mutual information. The function calculates the average mutual information (AMI) of the input time series and a copy of it that is time delayed by an amount T data points. The AMI is calculated for a range of time delays. The time delay that is returned to the main program to be used for the time delay embedding is that corresponding to the first minimum of the AMI.

```
% function definition
function[Tmin] = AMI(TimeSeries,bins)
% inputs
Tms = TimeSeries;
% check that Tms is a column vector. If not then transpose
dims = size(Tms);
if dims(2)~=1; Tms = Tms'; end
% length of sample
l = length(Tms);
% AMI for increasing T
for T = 1:500
    % time series
    s = Tms( 1 : (l-T) );
    q = Tms( (1+T) : l );
    % number of bins for the
```

```
16      % joint probability distribution
17      bins_j = sqrt(bins);
18      % probability distribution for the time series s
19      prob_s = hist( s, bins );
20      prob_s = prob_s/sum(prob_s);
21      % probability distribution for the time series q
22      prob_q = hist( q, bins );
23      prob_q = prob_q/sum(prob_q);
24      % joint probability distribution
25      prob_sq = hist3([s q], [bins_j bins_j]);
26      prob_sq = prob_sq/sum(sum(prob_sq));
27      prob_sq = prob_sq';
28      % AMI
29      summation = 0;
30      for i = 1:bins
31          if (prob_sq(i) == 0) || (prob_s(i)*prob_q(i) == 0)
32              summation = summation + 0;
33          else
34              summation = summation + ...
35                          prob_sq(i)* ...
36                  log2(prob_sq(i)/(prob_s(i)*prob_q(i)));
37          end
38      end
39      I_array(1,T) = abs(summation); % store AMI
40      T_array(1,T) = T; % store corresponding T
41  end
42  % T corresponding to first minimum of AMI
43  Idiff = diff(I_array);
44  count = 1;
45  while Idiff(count+1)/Idiff(count)>0
46      Tmin = T_array(count);
47      count = count + 1;
48  end
```

A.2.2 Function FNN

The following function determines the embedding dimension d for the time delay embedding of a time series using the method of false nearest neighbours. The function calculates the coordinates of the phase points that reconstruct the trajectory of the system in phase space for a range of embedding dimensions d and the time delay T that is determined beforehand by the function AMI in section A.2.1. The function then determines at which d all of the coordinates of the points in phase space stop changing and returns this d to the main program to be used for the time delay embedding. The coordinates of the points in phase space change as the dimension of the space is gradually increased until the dimensionality of the system becomes equal or smaller than the dimensions of the space in which it is being embedded. For example, the trajectory of the simple harmonic oscillator is an ellipse in two-dimensional space. An embed-

ding in one dimension would give phase points arranged along a line. Increasing the embedding to two dimensions would result in the coordinates of the phase points changing as they will be rearranged to form an ellipse in two dimensions. Increasing the embedding dimension further, to three dimensions for example, will not result in the coordinates of the phase points changing again since an ellipse in two dimensions remains an ellipse in three dimensions.

```
% function definition
function [D] = FNN(TimeSeries,TimeDelay)
% change of variables for convenience of coding
s = TimeSeries;
T = TimeDelay;
% length of input time series
L = length(TimeSeries);
% mean of observations
s_mean = sum(s)/L;
% approximate size of attractor
R_A = sqrt( sum( (s-s_mean).^2 ) /L );
% maximum embedding dimension
d_max = 20;
% percentage of false nearest neighbors 'PFNN' for
% increasing embedding dimension
for d = 2:d_max
    % generate matrix y
    % each column of y is a position vector
    % of a point in phase space
    for T_coeff = 0:(d-1)
        y((T_coeff+1),:)= ...
                    s((1+T_coeff*T):(L-(d-1)*T)+(T_coeff*T));
    end
    % length of delayed time series 'y'
    l = length(y);
    % find nearest neighbors y_NN and calculate R_d
    % create a vector array of ones,
    % for use in the 'for' loop that follows,
    % that has the same length 'l'
    % as the delayed time series 'y'
    clone = ones(1,l);
    for j = 1:l
        E = sqrt(sum((y(1:(d-1),j)*clone-y(1:(d-1),:)).^2));
        E(j) = NaN;
        [E_min, index] = min(E);
        R_d(1,j) = E_min;
        y_NN(1,j) = index;
    end
    % Calculate Change = R_{d+1}^2 - R_d^2
    for j = 1:l
    Change(1,j) = y(d,j) - y(d,y_NN(1,j));
    end
    % Calculate R_d_PlusOne
    R_d_PlusOne = sqrt(R_d.^2 + Change.^2);
```

```
    % Define threshold R_T
    R_T = 10;
    % Initialize counter for False neighbors
    False = 0;
    for j = 1:1
        if (abs(Change(1,j))/R_d(1,j)>R_T) || ...
           (R_d_PlusOne(1,j)/R_A>2) || ...
           (R_d_PlusOne(1,j)/R_A==2)
           %
           False = False + 1;
        end
    end
    PFNN_array(1,d-1) = (False/1)*100;
    d_array(1,d-1) = d-1;
    % Clear arrays for next loop
    clear y clone E R_d y_NN Change;
end
% Embedding dimension
[PFNNmin,D] = min(PFNN_array);
```

A.2.3 Function ODEpar

The following function calculates the model parameters *theta*1 and *theta*2, which are the case-specific coefficients in the model equations. The function receives the position *x_dns* and amplitude *T_dns* of the wave from the main program. These are the position and amplitude of the wave obtained from the direct numerical simulation in the case of the current work. The rows of these arrays contain the position and amplitude with respect to the wall-normal coordinate and the columns contain the position and amplitude with respect to time. For a wall-normal coordinate (row of *x_dns* and *T_dns*) at a time, the function calculates the Jacobian derivatives that relate the position and amplitude to their time derivatives. Finally, the function makes a least squares fit to the data points. The regression constants are the required values of *theta*1 and *theta*2. The process is repeated for every wall-normal coordinate (every row of *x_dns* and *T_dns*) to generate *theta*1 and *theta*2 vector arrays that contain the coefficients of the model equations as a function of the wall-normal coordinate.

```
% function definition
function [theta1,theta2,bestfit,akaike_weights] = ...
    ODEpar( t,Dt,NY,j_ignore,y_effect,x_dns,T_dns,steps,h)
%
t = t';
jnonzero = 1;
for jslice = 1 : NY
    if j_ignore(jslice) ~= 1
        % DIFFERENCING
        %%%%
```

```
        % Tmid
        T1 = T_dns(2:end,jslice);
        T0 = T_dns(1:end-1,jslice);
        Tmid = (T1 + T0)/2;
        % xmid
        x1 = x_dns(2:end,jslice);
        x0 = x_dns(1:end-1,jslice);
        xmid = (x1 + x0)/2;
        % DT
        DT = T1-T0;
        DT = DT';
        % Dx
        Dx = x1-x0;
        Dx = Dx';
        % T_dot
        DTDt = DT ./ Dt;
        % x_dot
        DxDt = Dx ./ Dt;
        % Tmid2
        T1 = Tmid(2:end);
        T0 = Tmid(1:end-1);
        Tmid2 = (T1 + T0)/2;
        % xmid2
        x1 = xmid(2:end);
        x0 = xmid(1:end-1);
        xmid2 = (x1 + x0)/2;
        % DTdot
        Tdot1 = DTDt(2:end);
        Tdot0 = DTDt(1:end-1);
        DTdot = Tdot1-Tdot0;
        % Dxdot
        xdot1 = DxDt(2:end);
        xdot0 = DxDt(1:end-1);
        Dxdot = xdot1-xdot0;
        % remove last element of DT and Dx so that their
        % length is the same as length of DTdot
        DT = DT(1:end-1);
        Dx = Dx(1:end-1);
        % DTdotDT
        DTdotDT = DTdot./DT;
        % DTdotDx
        DTdotDx = DTdot./Dx;
        % DxdotDT
        DxdotDT = Dxdot./DT;
        % DxdotDx
        DxdotDx = Dxdot./Dx;
        %%%%
```

```
63              f = fit(t(:),x_dns(:,jslice),'poly1');
64          else
65              % use only first 3 time steps
66              f = fit(t(1:3),x_dns(1:3,jslice),'poly1');
67          end
68          c = coeffvalues(f);
69          theta1(jnonzero) = c(1);
70          %%%%
```

```
                        aic_min = aic(i);
                    else
                        % check if next aic is
                        % smaller than initial aic
                        if aic(i)<aic_min
                            aic_min = aic(i);
                        end
                    end
                end
                delta_aic = aic - aic_min;
                % akaike weights
                akaike_weights(jnonzero,:) = ...
                                exp(-0.5.*delta_aic)./...
                                  sum(exp(-0.5.*delta_aic));
            end
        else
            akaike_weights(jnonzero,:) = [0 0 1];
        end
        % best fit
        [max_w,bestfit(jnonzero,1)] = ...
                        max(akaike_weights(jnonzero,:));

        % theta2
        theta2(jnonzero) = alpha(bestfit(jnonzero,1));
        %
        jnonzero = jnonzero + 1;
    end
end % end of for loop - jslices
```

A.2.4 Function ODEeqn

The following function defines the model equations in accordance with the argument description of the MATLAB function ode113, which is used in the main program to solve the equations numerically. Note that the analytic solution of the equations could be used instead.

```
% ODE system
% function definition
function dydt = ODEeqn(c,y,theta1,theta2,bestfit)
dydt = zeros(2,1);
dydt(1) = theta1;
if bestfit==1
    dydt(2) = theta2*y(2)^0.5;
elseif bestfit==2
    dydt(2) = theta2*y(2)^1;
elseif bestfit==3
    dydt(2) = theta2*y(2)^2;
end
```

Appendix B

Code blocks added to source files of the DNS flow solver BOFFIN

The following blocks of FORTRAN code are added to the source files of the DNS flow solver with the acronym BOFFIN (Boundary Fitted Flow INtegrator). This flow solver, developed at Imperial College London, is designed for LES/DNS simulation of reactive and non-reactive flow. This flow solver is used in the current work for the simulation of an advecting entropy wave in compressible, turbulent channel flow for the cases of adiabatic and convectively cooled walls. The original version of the flow solver does not have the built-in capability to add an entropy wave, that is an externally forced temperature fluctuation, to the flow. Furthermore, a convective thermal boundary condition that would enable adiabatic and convectively cooled conditions at the walls could not be defined in the original code. Hence, the code blocks in sections B.1 and B.2 are necessary for the simulations of the current work to be carried out. It should also be noted that in the original code the density field is coupled with the chemistry. As the simulation of the current work is of a non-reacting flow, that is the chemical reactions are not simulated, the density is decoupled from the chemistry and obeys the ideal gas law.

B.1 Entropy wave generation

The following code block is added to the main program script src/main_les.f90. The code generates a temperature perturbation in a plane cross-section of the flow. The temporal profile of the perturbation amplitude is a Gaussian function.

```
1 ! add this code block to boffin/src/main_les.f90 after the
2 ! solve u,v,w,P loop
3 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
4 !Tpulse for time step using the Gaussian function
5 Tpulse = Tamb + amp_wave*Tamb* &
6 exp(-0.5*( &
```

```
          ((Umean_bulk/(Ly))*(time-time0)-mu_wave) / &
          sigma_wave &
          )**2)
!should the wave be added?
if ((restart).and. &
                (waveInFlow).and. &
                                  (Tpulse.gt.Tamb+0.0005)) then
!smallest partition ID in which the wave lies
IDmin_wave=waveIglobal/(L-1)
IDmin_wave=floor(IDmin_wave)
do counter=0,NPJ*NPK-1
   IDmin_wave_array(counter)=IDmin_wave
end do
!partition IDs at inlet
do counter=0,NPJ*NPK-1
   IDs_inlet(counter)=counter*NPI
end do
!partition IDs at wave insert point
do counter=0,NPJ*NPK-1
IDs_wave(counter)=IDs_inlet(counter) + &
                                  IDmin_wave_array(counter)
end do
!build wave
waveI=waveIglobal-(IDmin_wave*(L-1))
do counter=0,NPJ*NPK-1
   if (ID.eq.IDs_wave(counter)) then
      build_wave=.true.
   else
      build_wave=.false.
   end if
   if (build_wave) then
      do J=1,M+1
        do K=1,N+1
         F(WaveI+JO(J)+KO(K)+nfo(nvf))=Tpulse
        end do
      end do
   endif
end do
end if
```

B.2 Convective thermal boundary condition

The following code block is added to the subroutine src/src_les/bndry_scalar.F to enable a convective thermal boundary condition to be defined at the channel walls. The wall temperature is no longer constant at the walls as in the original code. Instead, the wall temperature is variable and the heat flux is constant. The heat conducted through the no-slip fluid layer at the wall is equal to the heat convected to/from an imaginary (not simulated) external flow. The thermal conductivity of the fluid is assumed to be a constant. The convective heat transfer coefficient is set to zero for adiabatic walls and non-zero for convective heat transfer conditions.

```
1  !The new variables const_Temp, T_infinity, h, k_fluid and
2  !Deltay_wall are declared in src/src_les/module_les.f90.
3  !The user needs to supply T_infinity, h, k_fluid in input/les
4  !and Deltay_wall is calc. in src/src_les/input_les.f90 as,
5  Deltay_wall=(Ly*ratiodrj**(NODES_T/2-1)*(1-ratiodrj)) / &
6                                        (1-ratiodrj**(NODES_Y/2))
7  !where Ly, ratiodrj and NODES_Y are taken from les/input and
8  !are the channel-half width, stretch factor in y-direction
9  !and the number of nodes in y-direction respectively.
10 !in src/src_les/bndry_scalar.F and /implicit_bndr.F
11 !for constant wall temperature at the wall = false
12 !substitute the following for Twall:
13 Twall = (k_fluid/(k_fluid+h*Deltay_wall))*T_IJK + &
14 (h*Deltay_wall/(k_fluid+h*Deltay_wall))*T_infinity
```

Appendix C

Results from supplementary simulations

C.1 Entropy wave advecting in laminar flow

An entropy wave has a larger residence time in a laminar than in a turbulent flow. However, this does not necessarily imply that the same perturbation will undergo stronger attenuation in a laminar flow. This is because in a laminar flow the entropy wave decays exclusively due to molecular diffusion of its thermal energy towards the surrounding flow, which is a much slower process than turbulent diffusion.

Snapshots of an entropy wave advecting in adiabatic fully-developed channel flow are shown in figure C.1a for the case of a turbulent flow with $Re_{bulk} = 5684$ ($Re_\tau = 180$) and in figures C.1b and C.1c for the case of a laminar flow with $Re_{bulk} = 1120$. As explained in section 2.2.5, the entropy perturbation is added to the flow by perturbing the temperature in a cross-section. The perturbations added to each case are shown above the snapshots in figure C.1a-c in plots of the perturbation amplitude with respect to the number of iterations. The same time step is used for the turbulent and laminar simulations. Hence, the perturbations added to the laminar and turbulent flows are directly comparable when plotted with respect to the number of iterations. The perturbation added to the laminar flow in figure C.1b is the same as that added to the turbulent flow in figure C.1a. The perturbation added to the laminar flow in figure C.1c has the same peak amplitude as the others but is added to the flow at a slower rate. The time instant to which a snapshot corresponds is shown in the top right corner of the snapshot and is expressed as a function of the mean residence time of the turbulent flow $\tau_t = (U_{bulk}/2\delta)$, where U_{bulk} is the bulk mean velocity and δ is the channel half-height. The snapshots of the laminar flow cases correspond to the same time instants as those of the turbulent flow case. Since the residence time of the laminar flow cases is much larger than that of the turbulent flow case, there is an additional snapshot in the laminar flow cases in figures C.1b and C.1c showing the state of the perturbation once it has reached the channel exit.

The first snapshot of the entropy wave at time $\tau = 0$ shows the wave once it is added to the flow. In the turbulent flow case in figure C.1a, the maximum perturbation amplitude at time

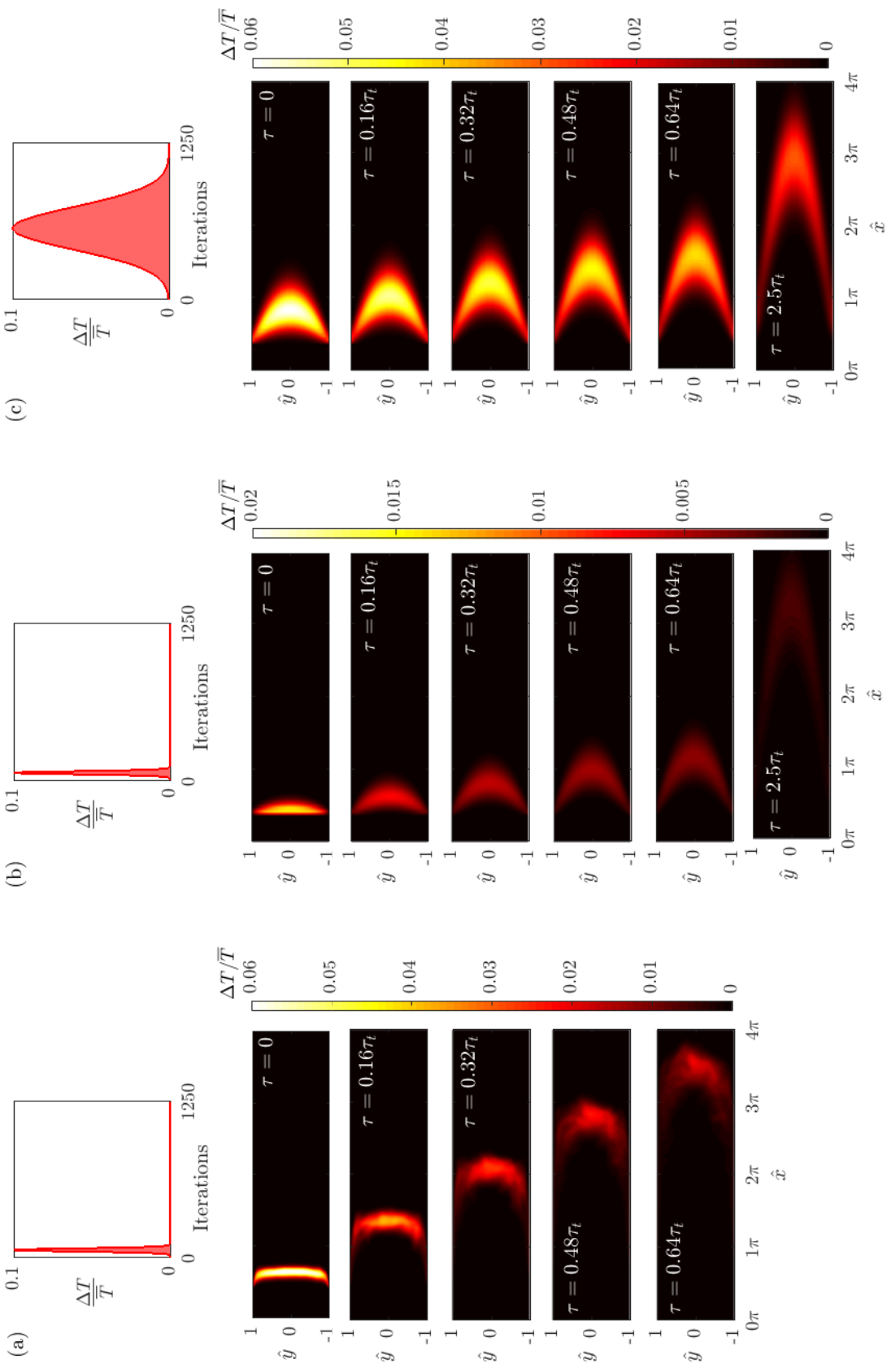

Figure C.1: Gaussian temperature perturbation advecting in fully developed channel flow with adiabatic walls: (a) turbulent flow, (b) laminar flow with the same perturbation as in the turbulent flow, and (c) laminar flow with a more slowly added perturbation.

$\tau = 0$ is $\Delta T/\overline{T} = 0.06$ and in the laminar flow case in figure C.1b it is $\Delta T/\overline{T} = 0.02$. The same perturbation undergoes stronger dissipation while it is being added to the laminar flow. This is because a stronger temperature gradient is driving the diffusion of the heat being added to the laminar flow. Since the flow velocity is much smaller in laminar flow than in turbulent flow, the volume of fluid effectively being heated as it passes through the perturbation insert plane is also much smaller. Consequently, the same amount of thermal energy is added to a smaller volume in the laminar flow resulting in a larger temperature gradient. By adding the perturbation to the laminar flow at a slower rate, which is what is done in the laminar flow case in figure C.1c, the temperature gradient at the perturbation insert plane is comparable to that in the turbulent flow. It is clear in figure C.1c that by doing so the maximum perturbation amplitude at time $\tau = 0$ is $\Delta T/\overline{T} = 0.06$, which is the same as in the turbulent flow case in figure C.1a.

The shape of the wave in the laminar flow cases in figures C.1b and C.1c is parabolic due to the laminar velocity profile. In the turbulent flow case in figure C.1a, the shape of the wave is mostly flat except near the walls where there is a large velocity gradient. The dispersion of the wave in regions of non-zero velocity gradient gives rise to temperature gradients in the wall-normal direction that enhance the transport of heat by molecular diffusion and thus, the dissipation of the wave amplitude by what is known as the shear dispersion mechanism. Since in turbulent flow the velocity gradient is extremely large close to the walls, dissipation of the wave amplitude by shear dispersion occurs mainly near the walls. Furthermore, turbulent fluctuations are also strongest close to the walls and hence, so is turbulent transport of heat. Since the shear dispersion and turbulent transport processes are both strong near the walls in turbulent flow, the amplitude of the wave in figure C.1a is dissipating faster near the walls. In laminar flow the velocity gradient changes gradually from the centreline to the walls and hence, the effects of shear dispersion are not more profound in a specific region of the flow.

The transport of heat by turbulent eddies in the case of the turbulent flow generates the rough amplitude distribution seen in the snapshots in figure C.1a. In contrast, the amplitude distribution in the laminar flow cases in figures C.1b and C.1c is smooth.

The maximum amplitude of the entropy wave in the turbulent case in figure C.1a decreases from $\Delta T/\overline{T} = 0.06$ at $\tau = 0$ to $\Delta T/\overline{T} = 0.02$ at time $\tau = 0.64\tau_t$, which is a 67% reduction in amplitude. In the same time interval, the maximum amplitude of the entropy wave in the laminar flow case in figure C.1b decreases from 0.02 to 0.005, which is a 75% reduction in amplitude. In the laminar flow case in figure C.1c, in the same time interval the maximum amplitude decreases from 0.06 to 0.04, which is a 33% reduction in amplitude. Therefore, the entropy wave in the laminar flow case in figure C.1b is dissipating faster than in the turbulent flow case and the entropy wave in the laminar flow case in figure C.1c is dissipating slower.

In conclusion, increasing the Reynolds number enhances the dissipation of the wave in the near wall regions. It is also observed that a high frequency perturbation in laminar flow undergoes strong dissipation by molecular diffusion due to the large temperature gradients that it

generates in the flow.

C.2 Entropy wave advecting in turbulent heat transferring flow — the effect of the wall heat transfer coefficient

In section 4.2.3, only the heat transferring case with $h = 200\,[Wm^{-2}K^{-1}]$ is shown. The effect of further increasing the heat transfer coefficient at the walls of the channel is shown in figure C.2, which shows the decay of the maximum temperatures in the streamwise cross-sections of the flow with respect to time t for various heat transfer coefficients. Increasing the heat transfer coefficient at the walls enhances the amplitude decay close to the walls and the edges of the wave are annihilated faster. As a result, the width of the wave in the wall-normal direction is smaller at larger heat transfer coefficients. In the adiabatic walls case, where $h = 0\,[Wm^{-2}K^{-1}]$, the wave extends all the way to the channel walls except during the late stages of its advection at time t_3 in figure C.2. In a channel that is long enough, the width of the wave could eventually become the same for all values of the heat transfer coefficient as seen at time t_3 in figure C.2. This happens because in a turbulent channel flow the shear dispersion and turbulent fluctuations that dissipate the wave in the adiabatic case are strongest near the walls, which is where the heat transfer mainly enhances the dissipation in the non-adiabatic cases.

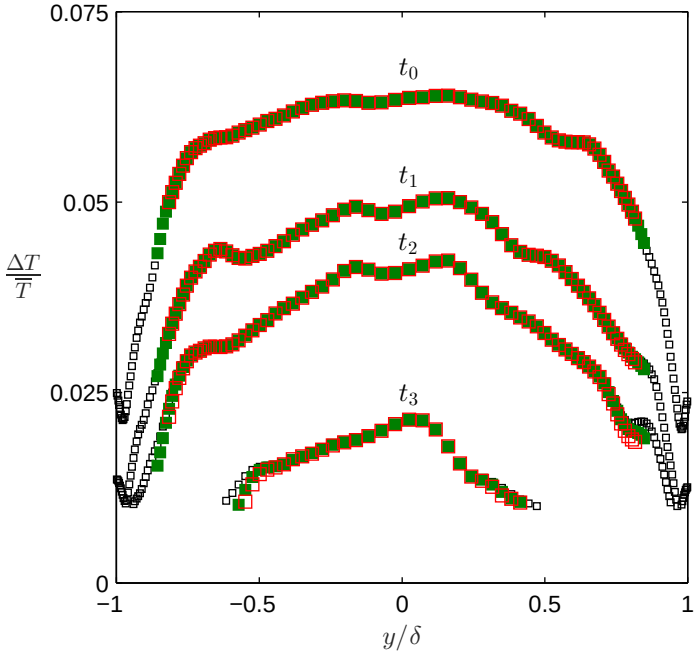

Figure C.2: Entropy wave amplitude decay in a fully developed channel flow with heat transferring walls: □ $h = 000\,Wm^{-2}K^{-1}$, ■ $h = 200\,Wm^{-2}K^{-1}$, □ $h = 800\,Wm^{-2}K^{-1}$